Call-A.P.P.L.E.™

$3.00 ($3.75 Canada, £2.00 UK)

Worlds Largest Apple User Group

January 1984

Earl Of Varspace

Page 9

HIGHLIGHT APPLE-CATIONS:

The Festive APPLE

Page 23

B. KEISTER

Apple PugetSound Program Library Exchange

Instant one-button color printing.

Press here.

It's just that easy! Any time you want to print what's on your Apple's screen just hit the copy button on your Transtar 315 color printer with our PICS card installed, and it's done! No special programming, no lengthy code sequences, no need to exit your program! Just press the button and it prints!

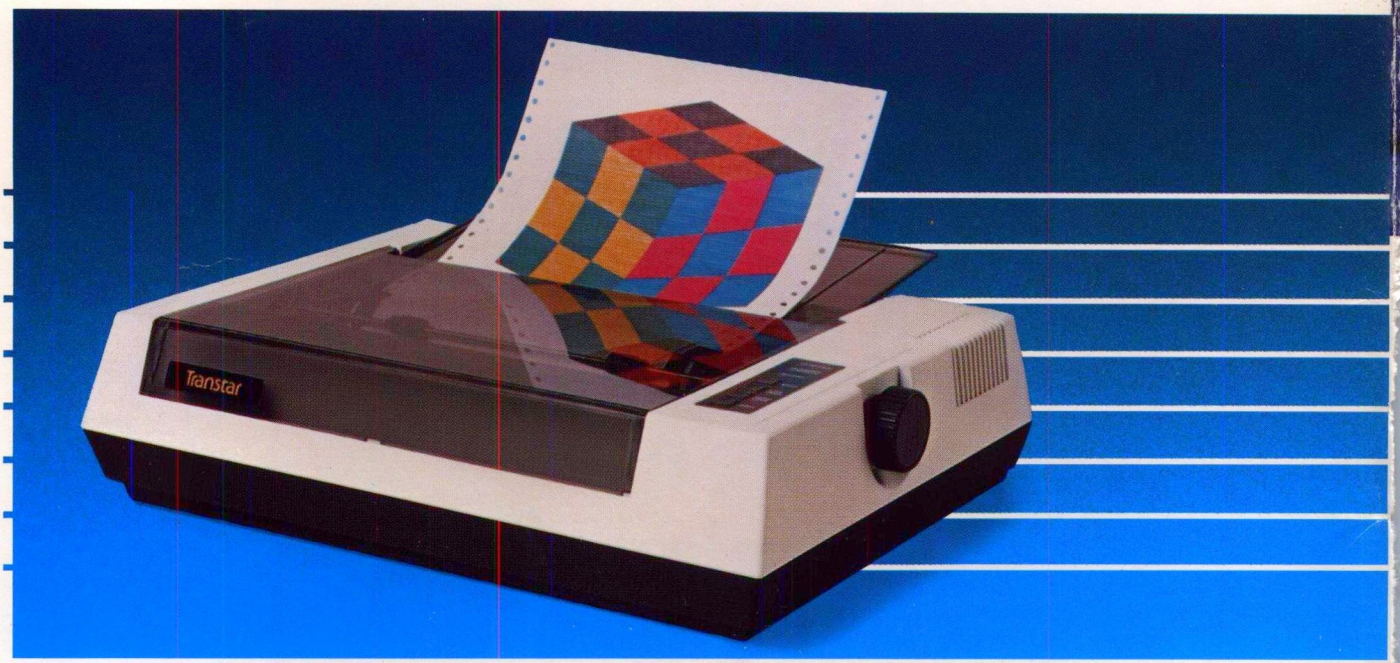
By adding the optional PICS card to your $599 Transtar 315 color printer, you've opened up a whole new world of easy color printing. For the first time ever, our PICS parallel interface card enables you to screendump virtually any program -- graphics, charts, games -- even copy-protected software! Specially designed only for the Apple II, II+, IIe, and Franklin computers, the Transtar 315 PICS card does the work of a parallel card and a lot more and costs only $119.95.

At the push of a button, Transtar's innovative new 4-color diagonal ribbon will print up to 7 colors and more than 30 shades in a single pass.

The 315 is precision-built to exacting standards by Seikosha, the most experienced company of the famous Seiko group--recognized worldwide for quality and dependability. In fact, one of the nicest things about Transtar's 6-month warranty on parts and labor is that you'll probably never use it!

Innovative, inexpensive, dependable, easy: the Transtar 315. Color printing has never looked so good!

Only $599.



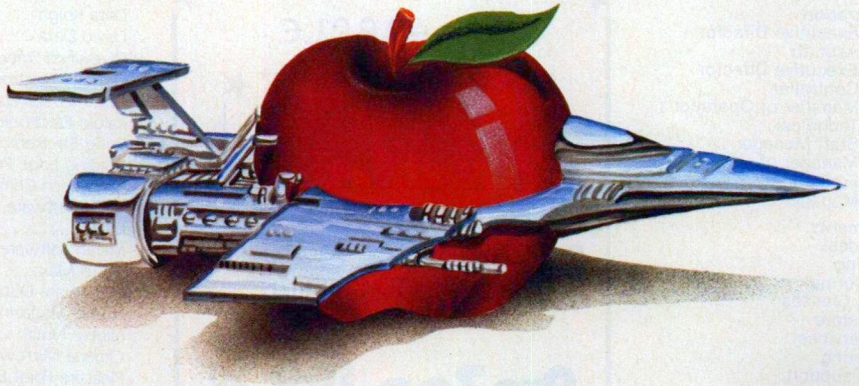
*PICS cards are currently available for Apples and Franklins. PICS cards for other computers will be available in the future.

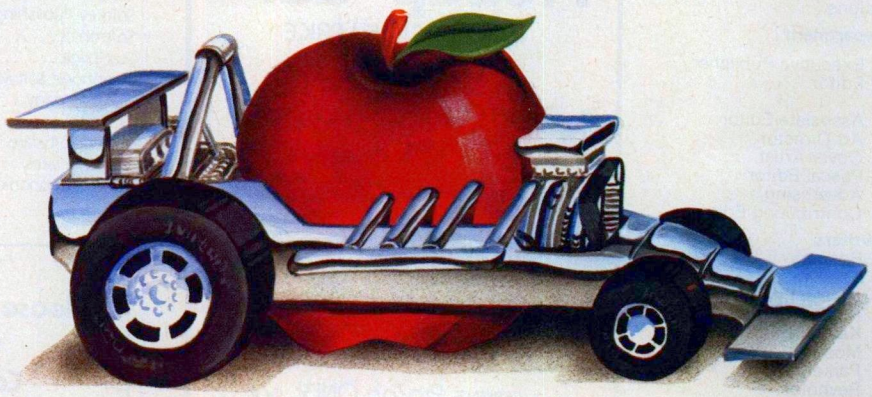
Transtar
A **Vivitar.** Computer Product

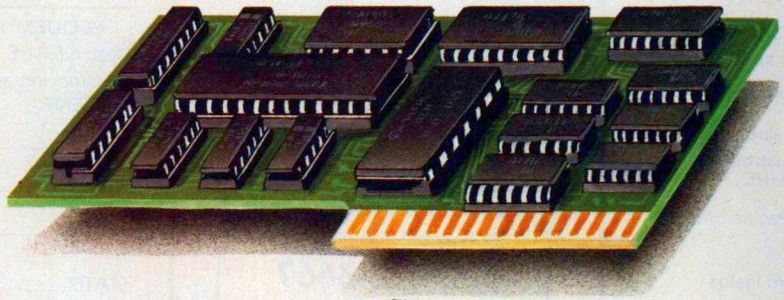
P.O. Box C-96975, Bellevue, WA 98009

Circle 26 on Reader Service Card

Three ways to Speed up your Apple II & Apple IIe for $295







The SpeeDemon™ from McT.

- SpeeDemon™ ● Makes any Apple II, II+, or IIe run 3½ times faster.
- SpeeDemon™ ● Makes your Applesoft, Apple Fortran, Word Processing, D.B. Master, Pascal, or Visicalc programs run up to 3½ times faster.
- SpeeDemon™ ● Costs less than any other speed up card.
- SpeeDemon™ ● Available immediately by mail. (We guarantee shipment within 7 days from receipt of order.)
- SpeeDemon™ ● Costs only $295.

Circle 17 on Reader Service Card

Please send me ____ SpeeDemons™ by return mail at $295 each. I have enclosed $ ____

I have an: ☐ Apple II ☐ Apple II plus ☐ Apple IIe This is for: ☐ Business Use

Or charge my: ☐ Visa ☐ Mastercard ☐ American Express ☐ Personal Use

My Acct.# is ____ Expires ____

Name ____

Address ____

City ____ State ____ Zip ____

Calif. Res. Add 6½% Sales Tax. Mail to: McT • 1745 21st Street • Santa Monica, CA 90404

M·c·T

A PRODUCT OF MICRO COMPUTER TECHNOLOGIES

Dealer Inquiries Invited

Apple is a registered trademark of Apple Computers, Inc. VisiCalc is a registered trademark of VisiCorp, Inc. DB Master is a registered trademark of Stoneware, Inc.

Apple PugetSound
Program Library Exchange
A.P.P.L.E. Building
21246 - 68th Ave. S.
Kent, WA 98032 - (206) 872-2245
Hours: 9am to 4pm PST, Mon-Fri.

Officers and Directors
Dick Hubert President
Bob Huelsdonk Vice-President
Dave Lingwood Secretary
Bob Clardy Treasurer
Ken Smith Director
Don Williams Director

Administration
Dick Hubert Executive Director
Don Steele Asst. to Executive Director
Dave O'Leary Controller
Sally Davies Manager of Operations
Steve Johnson Technical Staff Manager
Charles Stillman Manager of Education Services
Kevin Donohoe Manager of Marketing

Departments
Order desk
Tracing
Member Information
Education, Classes
Retail store
Dealer services
Marketing
Technical support
Testing and Evaluation
Operations

Editorial Department
Gary Kruger Executive Publisher
Val J. Golding Editor
Kathryn Hallgrimson Suther Associate Editor
Ed Adams Art Director
Ron Keister Cover Artist
Paul Sleeper Pascal Editor
Grawin Publications Advertising
Jonathan Post Contributing Editor

Staff writers
Cornelis Bongers
Wes Felty
Jim Hockenhull
Clifton M. Howard, M.D.
Ken Kashmarek
Paul Mosher
Bill Parker
Lee Reynolds
Clay Ruth
Michael Shadick
David G. Sparks
Rick Sutcliffe
Ralph H. Swerdlow, M.D.

Member Services

24 hour toll-free order line:
1-800-426-3667
(except Washington state)
Orders ONLY
Information not available

Programming/Technical Assistance
Call -A.P.P.L.E. HOTLINE
(206) 872-9004
9am to 3pm - 6pm to 10pm (Pacific time)
7 days per week

Consultants
see listing on tear-off card inside

A.P.P.L.E. Crate Bulletin Board
(206) 872-6789

Meetings
7:30pm third Tuesday of each month
A.P.P.L.E. Office
21246 68th Ave. S., Kent
(except as noted)

Membership Information
is available on postpaid tear-off card inside

A.P.P.L.E. on the SOURCE
A special section of The Source is open only to A.P.P.L.E. members who may download Call -A.P.P.L.E. programs. To access, type: CALL -APPLE from Source command mode. To register, send mail to ST1570 with your name, membership number, expiration date and Source account number.

The January meeting will be held one week late on January 24, 1984.

Advertising inquiries please contact:
GRAWIN PUBLICATIONS
1020 Lloyd Building, Seattle, WA 98101
(206) 223-0861

Call —A.P.P.L.E.

MONTHLY MEMBER SPECIAL

- Be the "first on your block" with a ProDOS utility.

ProZap $22.50
INTRODUCTORY PRICE
Product Code — SPZAP

- Reads, edits and writes any ProDOS block or DOS 3.3 sector.
- Includes PRODOS. SYSTEM and BASIC. SYSTEM as SYS files.

Note: Included documentation covers ProZap ONLY, not ProDOS.

- On the same disk: Monitor utility examines, searches memory for hex, ASCII.

Offer good til 3/31/84

shipping weight 2 oz

24 hour toll-free order no.
800-426-3667
(except in Washington)
Washington state residents add 8.1% sales tax

A.P.P.L.E. Orders
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

AD INDEX

A.P.P.L.E. 2, 80
Access Micro 73
Alpine Computer Systems 51
Apple Computer 40, 41
Beagle Brothers 52, 53
Bytes & Pieces 55
Central Point Software 32
Computer Software Associates 21
Control Data Corp 4
Data Knight 60
David Data 56
Diversified Software Research 60
Douglas Electronics 22
Gourmet Software 73
Kalglo Electronics Co. Inc. 54
Laing Electronics 8,9
Leading Edge Products ..... BC
Macrotech Computer Products 44, 45
Manx Software 73
McJoy 1
Merlin Software 55
Micro Max 56
Microware Distributing 5
Nibble Magazine 76
Nibble Notch 56
Omega Microware 64
Pleasure Point Enterprises 54
Quantum Leap International 48
Softkey Publishing 46
Softronics 8
Star Logic 69
Telephone Software Connection 59
3M Company ..... IBC
Transtar ..... IFC
Trutec Software 46
VF Associates 59
Zoom Telephonics 6

Please tell our advertisers you saw it in Call —A.P.P.L.E.

REQUEST FOR CHANGE OF ADDRESS

Dear A.P.P.L.E.
I am moving and my present address is:
PRESENT:

NAME _____
ADDRESS _____
CITY _____
STATE _____ ZIP CODE _____
EFFECTIVE DATE _____
NEW ADDRESS:
NAME _____
ADDRESS _____
CITY _____
STATE _____ ZIP CODE _____
MEMBERSHIP NUMBER _____

* Please fill out and return this notice six weeks before moving.

Apple user groups may reprint without prior permission any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commerical applications of programs herein, the author should be contacted c/o General Office. Call —A.P.P.L.E. is published monthly by Apple Puget Sound Program Library Exchange, 21246 - 68th Ave. S., Kent, WA 98032. Subscriptions are not available but copies are mailed free as a benefit of A.P.P.L.E. membership.

Call-A.P.P.L.E.™

Volume VII, Number 1 January 1984/Menu

Val. J. Golding, Editor
Kathryn Hallgrimson Suther, Associate Editor

ENTRÉES

9 FROM THE VERY CORE OF APPLE:
Earl of Varspace
Cornelis Bongers/Call —A.P.P.L.E. staff writer

23 **Highlight Feature Section**
SPECIAL APPLICATIONS:
The Festive Apple
Burt Webb

27 SPECIAL APPLICATIONS:
Computerizing Science Exhibits
Dave Taylor

30 SPECIAL APPLICATIONS:
Higher Text In Action
Steve Brugger

33 A CALL —A.P.P.L.E. LANGUAGE REVIEW:
"C" What I Mean? (Part Two)
Jim Hockenhull/Call —A.P.P.L.E. staff writer

À LA CARTE

37 **Business Feature Section**
THE SPREADSHEET SHEET:
Data Analysis (Part 2)
Jeffrey Jacques, Ph.D.

43 **Strictly Business**
Mike Regeimbal

47 **Education Feature Section**
Flying High. . . With E-Z Pilot
Earl Keyser

49 A CALL —A.P.P.L.E. BOOK REVIEW:
Mind Storms
Eric Flescher/reviewer

50 A CALL —A.P.P.L.E. SOFTWARE REVIEW:
Facemaker
Garry Margolis/reviewer

55 **Telecommunications Conference**
Michael Thyng

57 **1983 Call —A.P.P.L.E. Index**
Kathryn Hallgrimson Suther/Val J. Golding

61 **Hardware Feature Section**
Koala Pad
Alan R. Fischer/reviewer

64 **The Gaming Room**
Val J. Golding

65 **Technical Feature Section**
ProDOS and Friends
Cecil Fretwell

70 NORTHERN SPY:
A Canadian Harvest
Rick Sutcliffe/Call —A.P.P.L.E. staff writer

79 **Poet-tree**
Michael Shadick/Sallie Stephenson

HORS d'OEUVRES

5 The Editor Bytes Back
the editor

7 Bytes From A.P.P.L.E.
Val J. Golding

51,59 So What Did You Expect?

74 **Letters Feature Section**
A.P.P.L.E. Doctor
Bruce F. Field

77 Write -A.P.P.L.E.
the readers

80 Unclassifieds

Entire contents copyright© 1984 by
Apple PugetSound Program Library Exchange

GIVE YOUR CHILD THE PLATO EDGE IN ALGEBRA.

For use with the Apple II Plus and Apple IIe:

New PLATO® lessons in Elementary Algebra*
Help your child feel confident about learning algebra skills. This new PLATO series helps simplify Exponents, Polynomials, Roots and Radicals, Factoring and other Algebra components. Practice problems change at random and examples of solutions help keep kids motivated.

New PLATO lessons in Computer Concepts†
This series helps kids understand the computer and lets them practice what they learn. Lessons include: The Computer Keyboard, Storage and Memory, Files and Editing and Databases.

Widen your child's world
Other PLATO lessons include Elementary Math, Foreign Languages, Physics-Elementary Mechanics, Computer Literacy and ‡Keyboarding.
All PLATO micro courseware is available for the Apple II Plus and Apple IIe. Selected lessons are available for the TI99/4A and Atari 800.

For a free catalog
See the growing line of PLATO micro courseware at selected retail outlets. For a free catalog, call toll-free: 800-233-3784. (In Calif., call 800-233-3785.) Or write: Control Data Publishing Co., P.O. Box 261127, San Diego, CA 92126.

*Developed with Courses by Computers, Inc.
†Developed with Continuous Learning Corporation.
‡Developed with Gregg/McGraw-Hill.

Warranty available free from Control Data Publishing Co., 4455 Eastgate Mall, San Diego, CA 92121

PLATO
COMPUTER-BASED EDUCATION

CONTROL DATA PUBLISHING

Circle 6 on Reader Service Card

THE EDITOR BYTES BACK

the editor

NINETEEN–eightyfour marks the beginning of **Call –A.P.P.L.E.'s** seventh year of publication as the leading journal by and for members of A.P.P.L.E., the world's largest Apple computer user group. What better way to bring 1984 to your attention than with our striking new cover, designed by **Ron Keister,** the noted illustrator. Ron has been commissioned to create some exciting covers for us, and you will be able to enjoy his work in each coming issue.

What's up, Doc? We are. We're taking a peek into the future, and our crystal ball tells us a new cover is not all you'll see happening in **Call –A.P.P.L.E. Gary Kruger** joins A.P.P.L.E. as Executive Publisher. Gary's responsibilities will include overall coordination of all publications, advertising and production. At the same time we would like to welcome **Ed Adams** as Art Director. In the months ahead, you will see Ed's fine touches on the pages of **Call –A.P.P.L.E.** and our other publications. Welcome aboard, gentlemen, it will be a pleasure working with you.

Still more pleasant changes are in store for you, but they can keep for a while as we introduce the lead story on our new cover.

After too long an absence, staff writer **Cornelis Bongers** is with us again. We have been keeping Cornelis busy editing another issue of **Call –A.P.P.L.E. In Depth,** due out later this year. **From the Very Core of Apple** brings us the **Earl of VarSpace,** a complete system of local variables in Applesoft, written as only Cornelis Bongers can. We know it will find immediate use in many of your applications.

Burt Webb (The Computer Home) heads off our **Highlight** section with **The Festive Apple,** a tale of how Apples were used to coordinate and manage events at Seattle's annual Folklife Festival. In the same section, **Dave Taylor** tells of **Computerizing Science Exhibits** at the Pacific Science Center, while **Higher Text in Action** is a followup to Dave's story, written by **Steve Brugger.**

Elsewhere in our pages, you will also find the concluding portion of **Jim Hockenhull's C What I Mean,** an in depth review of the Aztec C compiler. In our companion column, **Bytes From A.P.P.L.E.,** we tell you about our new ProDOS utility, **ProZap** and between these pages we offer **Cecil Fretwell's ProDOS and Friends,** a further look at Apple Computer's new disk operating system. (Readers may wish to brush up and re-read **Taking the Pro Out of ProDOS** in our November issue.)

Something that should not be overlooked in this issue is our **1982 Call –A.P.P.L.E. Index.** This is your key to adventure in the world of Apple, A.D. 1982. Game addicts are warned to skip **The Gaming Room,** unless you're prepared to pay $$$ for some of the neat games reviewed in this month's column.

Educators will not want to miss our education section, which this month includes another foray into the realm of the Pilot language, skillfully piloted by **Earl Keyser** with **Flying High.** We've at last reached the end of our rope, if not wit, yet there is still more to be seen. Just flip through the pages for your favorite features.

Ciao

Guest Speaker

Randy Wigginton, co-author of Applesoft and DOS 3.3 will be the guest speaker at our meeting Tuesday, January 24, 1983 which has been moved to the fourth Tuesday for this occasion. We can't say what Randy's topic will be, but just keep in mind Apple Computers announcements of new products scheduled for the first quarter of 1984.

FOR APPLE II PLUS, FRANKLIN, APPLE IIe

Font Down Loader

Expand the capacity of your printer hundreds of times

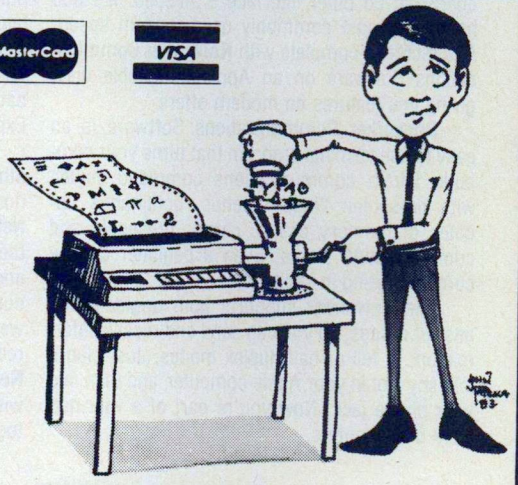
Load custom fonts into your Apple® Matrix Printer, Prowriter™ 8510A, OKI® Microline 92, 93, 84 Step II, and Epson® FX and use them with virtually every word processor to turn your printer into a **custom typesetter.** After the fonts are loaded, they will stay in your printer until it's turned off. A font editor is also provided to allow you to create your own graphics, text, foreign language letters, math and electronics symbols to load into your printer. On-Disk (Specify Printer) $39.00

New improved versions with drivers for Grappler, Pkaso, Wizard and most other intelligent parallel boards.

$100 REWARD

Submit the best or most unique font using the above software and we will make you $100 richer. Other prizes for the first 25 runners up.

MasterCard VISA



"There's got to be a better way to load fonts!"

MICRO WARE
P.O. Box 113
Pompton Plains, N.J.
07444

Dealer and Distributor Inquiries Invited

CALL (201) 838-9027

Circle 19 on Reader Service Card

APPLE® MODEM. WITH EVERYTHING.

Now including a valuable subscription to THE SOURCE℠ AMERICA'S INFORMATION UTILITY

$129

THE NETWORKER™ MODEM

Your best buy in modem history. The **Networker**™, a plug-in single-slot direct connect modem for the Apple II family of computers. Send electronic mail to a friend or business associate, use your school's computer, access hundreds of computer bulletin boards or thousands of data bases for up-to-the-minute news, sports, weather, airline, and stock information.

There's absolutely nothing else to buy. You get the modem board, communications software, and a valuable subscription to America's premier information service, THE SOURCE℠. For $129 it's an unbeatable value.

This is the modem that does it all, and does it for less. The Apple Communications Card is on board, so no other interface is needed. It's 300 baud, the most commonly used modem speed. And it comes complete with **Networker** Communications Software on an Apple-compatible disk, giving you features **no** modem offers.

Networker Communications Software is an easy to use terminal program that turns your computer into a communications command center, with on-screen "help" menus, continuous updates of memory usage, carrier presence, and communication status. Even a beginner can be communicating in seconds.

The **Networker** supports both originate and answer modes, so you can send and receive information, in full or half duplex modes. Just plug it into any slot in your Apple computer, and then into your phone jack. Now you're part of a vast network of computers.

NETMASTER™ COMMUNICATIONS SOFTWARE

For $179 we include with the **Networker** our **Netmaster** Communications Software for advanced users. **Netmaster** will let you transfer games, computer graphics, programs, sales reports, documents—in fact, any Apple file of any size—to another computer, directly from disk to disk, without errors, even through noisy phone lines.

For transfering information between computers, **Netmaster's** superb error checking and high speed are an unbeatable combination. With a **Netmaster** on each end, you can transfer information three to five times faster than other communications packages like Visiterm™ or ASCII Express™. Error free.

But **Netmaster's** not stuffy. It will talk to those other communications packages. It's just that they don't work as fast and they don't check errors like **Netmaster.** And **Netmaster** doesn't only work with the **Networker** modem. Even if you already have another modem for your Apple, **Netmaster** is an outstanding value in communications software, so we sell **Netmaster** by itself for $79. **Netmaster** requires 48k of RAM, one disk drive, and the **Networker** or another modem. It can also be used without a modem when two Apples are hard-wired together.

THE SOURCE℠

Your purchase of the **Networker** with or without **Netmaster** comes complete with a membership to THE SOURCE, with its normal registration fee fully waived. THE SOURCE will put a world of electronic information and communication services at your fingertips—instantly. Electronic mail and computer conferencing. Current news and sports. Valuable business and financial information. Travel services. A wealth of information about personal computing. Even games. All fully compatible with your equipment, and ready to use at once.

Networker, Netmaster, and **Networker/Netmaster Combo** are all produced in the USA by ZOOM Telephonics, producers of the popular **DEMON DIALER.** The Networker is FCC registered, and all products come with a one-year warranty.

HOW DO I GET ONE? OR TWO?

Ask your computer dealer, or call one of our sales offices listed below for the dealer nearest you. Or call us direct at 1-800-631-3116, and we'll tell you where to get one.
(In Massachusetts, call 1-617-423-1288.)

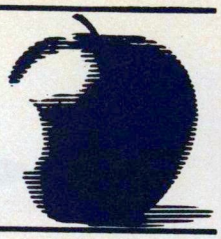
SALES OFFICES

Boston, MA (617) 329-4292; **Metro New York** (914) 834-5555; **Philadelphia, PA** (215) 638-4350; **Pittsburgh, PA** (412) 276-3310; **Washington, D.C.** (301) 942-4900; **Atlanta, GA** (404) 876-2288; **Miami, FL** (305) 945-8844; **Rochester, NY** (716) 385-6266; **Dayton, OH** (513) 890-5159; **Cleveland, OH** (216) 779-4435; **Chicago, IL** (312) 234-5911; **Minneapolis, MN** (612) 944-8640; **Lenexa, KS** (913) 541-1411; **Dallas, TX** (214) 747-0081; **Denver, CO** (303) 744-2323; **Salt Lake City, UT** (801) 486-2346; **Phoenix, AZ** (602) 265-9201; **Los Angeles, CA** (213) 949-9405; **San Francisco, CA** (415) 579-6561; **Portland, OR** (503) 292-3585; **Redmond, WA** (206) 881-8778.

ZOOM Telephonics, Inc.
207 South Street, Boston, MA 02111

The following are trademarks: Apple by Apple Computer, the Source servicemark by Source Telecomputing, Visiterm by Visicorp, ASCII Express by Southwestern Data Systems, DEMON DIALER by ZOOM Telephonics. © 1983 ZOOM Telephonics, Inc.

BYTES FROM A.P.P.L.E.



Val J. Golding

PRODOS is the key word, the hot item for January, with Apple Computer expected to announce its availability soon, and A.P.P.L.E., as expected, is on top of the situation. With the introduction of ProDOS, it is anticipated many of our members will be interested in heading off to the wild blue yonder, exploring well in advance, many of the ProDOS details we could bring you in print at this date.

ProDOS is a brand new disk operating system for the Apple][and Apple *IIe* that, in addition to many other innovations, offers complete compatibility with the Apple *III* and hard disk systems. So far, we have covered it with an introductory article in November by *Gary Charpentier* and *David Sparks,* with an additional article in this issue by *Cecil Fretwell.* More are on the way.

What this brings us down to is our new utility by *Gary Charpentier,* **ProZap. ProZap** uses intrinsic ProDOS routines to read, edit and write any ProDOS block or DOS 3.3 sector, all from a ProDOS environment! Also included on the diskette is a utility to dump to disk, printer or monitor, a hex or hex/ASCII listing or disassembly, plus a search routine for hex or ASCII bytes.

Between now and the end of March, you can order **ProZap** at the introductory price of $22.50. Thereafter it will be $25.00. The order code is SPZAP. Shipping date is expected to be February 1st, but because we can never be certain of the exact availability date of a new product, in order to save the $2.50, you may not include any other items on the same order form or telephone order. This saves us the possibility of splitting an order to prepare a separate back order form. Fair enough?

We suspect ProDOS is going to be pretty big in months to come, and we have taken this in account in looking at our list of potential new products and revisions to existing products. One hint we can drop is to look for **Apmail.Pro,** a version of our Apmail 3.0 mailing list with extensive improvements and of course running under ProDOS. With a bit of luck, you may find an announcement in the next issue. Lots of other things are in the works as well, but you'll just have to hold your breath.

From time to time, members have inquired about our producing a monthly disk of *Call –A.P.P.L.E.* programs. Currently we release an anthology diskette every three months, which contains files of programs published in *Call –A.P.P.L.E.* for the period specified. In some cases this has not been sufficient, since many of you do not want to wait two or three months to save typing a program in. The $12.50 that we charge for the **Anthology** series actually does not cover our cost of production, thus we have not so far been anxious to produce anything at the same price level 12 times per year, as opposed to four times per year. The only other alternative available to members is that those who own modems and subscribe to *The Source* may download our published programs.

In searching for a resolution to this problem, yet offering a monthly diskette, we arrived at a potential solution. It requires your understanding and participation to make it feasible, and it is necessary to establish a few strict guidelines. We have decided to proceed on a trial basis to see what member response is. We need to have a minimum of 100 prepaid orders to continue with this project. The essential conditions are outlined in the next paragraph. If you decide to order, any funds received will be set aside in a special fund and in the event we do not proceed, will be taken up as a credit to your A.P.P.L.E. account. Cash refunds will **not** be processed.

Two options are available: A six month advance subscription to a *Call –A.P.P.L.E.* disk of the month, or: A 12 month advance subscription to a *Call –A.P.P.L.E.* disk of the month. Only two anniversary dates will be recognized, January and July. This is absolutely essential to keep our costs down and provide you with diskettes at less than $10 each. Diskettes will be shipped by first class mail or UPS and will reach you on approximately the first of the month of the date of issue, perhaps in advance of your copy of *Call –A.P.P.L.E.* Orders placed now, if we proceed, will commence with programs published in the January, 1984 issue.

A six month subscription will cost $55, and the order code is SCA6. This amounts to $9.17 per disk. You can save more on a 12 month subscription at $95, which breaks down to $7.92 per diskette. The order code is SCA12. To help us in establishing this new member service, we ask for your early response and to include no other order in the same envelope or phone order.

So here you have it. Please let us know at your earliest opportunity if we have come up with a product that appeals to you. Our interest is in serving you, and providing products at affordable prices. The next step is yours.

Special Notice

Raising prices is never a pleasant topic, but it is one we have to face. We have been able to maintain out $20 annual dues rate for several years, notwithstanding many new and improved member services. We are a non–profit corporation, and while we exist to serve our members, it does not mean that we should do so at a loss.

As of January 1, 1984, our new annual dues rate will be $26. Much of the increased revenue will go to improving, broadening and expanding *Call –A.P.P.L.E.* Some of what is in store for you, you have seen creep in over 1983. Our new cover this month serves as only a hint of what is to come, thus we ask your understanding when we ask for the extra bucks. *Call –A.P.P.L.E.* is and will continue to be the premium publication our members have indicated they want to see.

DOS, CP/M,® PASCAL

The only communications software you'll ever need!

Softerm 1
The Complete, Upgradeable Package for Home or Business Use

Softerm 1 is a powerful and flexible terminal emulation program that operates on an Apple® II, II Plus, or IIe to provide basic terminal communications to a variety of host computers, timesharing services, and information services such as *The Source*℠, *CompuServe*® *and the Dow Jones News/Retrieval*®. It operates full or half-duplex at speeds up to 9600 bps using either a direct connection or any standard manual or auto-dial modem. Features include user-defined keyboard macros, built-in phone book for automatic dialing, terminal mode line capture simultaneously to print or disk, copy screen to print or disk, and terminal status display.

DOS, CP/M, and PASCAL File Compatibility Combined In a Single Program

Softerm 1 incorporates an advanced file manager which provides compatibility with DOS 3.3, CP/M, and Pascal disk formats for all file operations including file transfers. And at speeds up to 5 times faster than standard Apple DOS! Built-in disk utilities provide *INIT, CATALOG, RENAME,* and *DELETE* commands for all disk formats. Wildcard match characters can be used whenever filenames are entered.

Local file transfers allow DOS, CP/M, or Pascal files to be displayed, printed, or even copied to another disk. For example, a file on a CP/M formatted disk in Drive 1 could be copied to a Pascal formatted disk in Drive 2 providing a complete format conversion capability. Numerous editing options such as tab expansion and removing unwanted characters allow easy reformatting of data to accommodate the variations in data formats used by host computers.

Multi-Protocol File Transfer Capability

Softerm 1 offers file transfer methods flexible enough to match any host computer requirement. These include the *character* protocol with user-definable characteristics to provide maximum flexibility for text file transfers to any computer. The CP/M User's Group standard *XMODEM* protocol may be used for binary file transfers with systems using the CP/M operating system. The intelligent *Softrans* protocol can be used to transfer *any* type file and provides automatic binary encoding and decoding, error detection and automatic retransmission, and data compression to enhance line utilization. A FORTRAN 77 source program is supplied with Softerm 1 which is easily adaptable to any host computer to allow communications with Softerm using the Softrans protocol. Specific host computer versions of the Softrans FORTRAN program are available on request.

Softerm file transfer utilizes an easy to use *command language* which may be executed interactively or from a *macro* command file which has been previously entered and saved on disk. Twenty-three high-level commands include *DIAL, CATALOG, SEND, RECEIVE, ONERR, MONITOR, HANGUP,* and others. A *SCHEDULE* command even allows file transfers at a specific date and time.

Softerm 2
The "Choice of Professionals"

Softerm 2 includes all features of Softerm 1 and provides an *exact* terminal emulation for a wide range of conversational and block mode CRT terminals. Special function keys, sophisticated editing features, even local printer capabilities of the terminals emulated are fully supported. In fact, your host computer won't know the difference! *All* of the following emulations are included in Softerm 2 and the list is growing...

ADDS Regent 20, 25, 40, 60 • ADDS Viewpoint • Data General D200 • Datapoint 3601 • DEC VT102, VT52 • Hazeltine 1400, 1410, 1500, 1520 • Honeywell VIP7205 • IBM 3101 Model 10 and 20 • Lear Siegler ADM-3A, ADM-5 • TeleVideo 910, 925

You'll Never Outgrow It

For the latest program enhancements, you can access the Softronics Online Update Service 24 hours a day, 7 days a week. New hardware support or terminal emulations are immediately available to all Softerm users.

Softerm 1—$135 Softerm 2—$195

Available now from your local dealer or Softronics, Inc.

*Apple is a registered trademark of Apple Computer, Inc., CP/M is a registered trademark of Digital Research, Inc., Dow Jones News/Retrieval is a registered trademark of Dow Jones, Inc., The Source is a service mark of Source Telecomputing Corporation, CompuServe is a registered trademark of CompuServe, Inc.

SOFTRONICS

6626 Prince Edward, Memphis, TN 38119, 901-683-6850

Circle 24 on Reader Service Card

A.P.P.L.E. CONSULTANTS

John A. Hall, Jr.	Apple //e (Eastern U.S.)	(617) 241-9050
Cecil Fretwell	Apple //e (MidWestern U.S.)	(319) 236-0961
Bob Huelsdonk	Apple /// (Western U.S.)	(206) 362-4910
Harry T. Hanson	Apple /// (Eastern U.S.)	(201) 467-0712
Carmen Picardi	Apple /// (Eastern U.S.)	(603) 895-4447
Stuart Yusem	Apple /// (Eastern U.S.)	(609) 822-9306
Tim Smith	Apple /// (MidWestern U.S.)	(389) 734-7451
Neil Clemmons	Agriculture	(314) 968-9618
Erv Edge	Assembly (Alaska)	(907) 337-7620
Dennis Baumgarte	Assembly (Eastern U.S.)	(914) 298-8756
Michael Thyng	Business	(206) 524-2744
Murray Arnow	CPM	
Steven E. Blair	Data Bases	(913) 782-5901
Jerry Rogan	Data Bases, Business	(319) 332-4166
William P. Steinberg	DOS (Eastern U.S.)	(212) 921-0663
Art Schumer	DOS (Northern U.S.)	(701) 282-7907
Lee Meador	DOS (Southern U.S.)	(817) 469-6019
WM. Reynolds III	DOS/Tech (Western U.S.)	(503) 472-4028
Ted Perry	Education	(916) 485-1690
Cornelis Bongers	European Consultant	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Jim Eatherly	Games	(202) 232-6046
Tedd Sperling	Geoscience	(517) 393-1296
Stan Axelrod	Hardware Design/I-O Control	(415) 563-1840
Tom Deal, M.D.	Handicapped Applications	(206) 634-2847
Randy Halvorson	Hardware Interfacing	(206) 783-9040
Ed So	Hi-Res	(408) 267-3448
Stowe Keller	Mathematics	(804) 973-7647
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Stephen F. Lee	Music	(704) 289-4102
Tom Woteki	Pascal (Eastern U.S.)	(202) 547-0984
Tim Smith	Pascal (Midwestern U.S.)	(309) 734-7451
Paul E. Sleeper	Pascal (Western U.S.)	(415) 494-1915
Bruce E. Sargent	Pascal (Western U.S.)	(206) 488-1161
Dennis R. Cohen	Pascal (Western U.S.)	(213) 956-8559
Earl Keyser	Pilot Language	(612) 429-5570
Randy Warrens	Pilot Language	(608) 837-9566
Hersch Pilloff	Stock Market	(301) 292-3100
Jim Hockenhull	Video Interfacing	(509) 334-4137
George L. Hefter	VisiCalc (East)	(617) 256-8148
Vince Aderente	VisiCalc (North)	(312) 289-9685
Dr. Bob Richard	VisiCalc (South)	(504) 466-0894
Brian Abrahams	VisiCalc (South Africa)	(Durban) 315442
Michael Ginsberg	Word Processing	(805) 688-1465

ORDER

PRODUCTS LISTED ON REVERSE

Membership # _____ Expiration date _____ Date _____

Name _____ Address _____

City _____ State/Country _____ Zip _____ Phone _____

PRODUCT		Unit Price	*FOREIGN SHIPPING WT.	AMOUNT
1				
2				
3				
4				
5				
6				

Mail to:
A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

VISA Account No. _____ Exp. Date _____

MasterCard Account No. _____ Exp. Date _____

Subtotal _____

**Washington sales tax _____

TOTAL AMOUNT _____

CALL TOLL-FREE
1-800-426-3667 ORDERS ONLY OUTSIDE WASHINGTON ONLY (24 Hours)

***FOREIGN MEMBERS**
****WASH STATE RESIDENTS ADD 8.1% SALES TAX.**

☐ APPLE // | //e PLEASE SPECIFY
☐ PLEASE USE PRODUCT NAME AND CODE WHEN ORDERING

FOREIGN MEMBERS: Please calculate postage based on weights in product descriptions.
(See postage table below for cost)

FOREIGN SHIPPING CHART

POSTAL TABLE		I West Indies Central American Columbia, Venezuela	II All South American Countries (except in Table I), Europe, N Africa	III Asia, Mid East, S. Africa Australia and Pacific Islands
0 oz	1 oz	$0.50	$0.60	$0.70
1	2	.76	.96	1.16
2	3	1.02	1.32	1.62
3	4	1.28	1.68	2.08
4	6	1.54	2.14	2.74
6	8	1.80	2.60	3.40
8	10	2.06	3.06	4.06
10	12	2.32	3.53	4.72
12	14	2.58	3.98	5.38
14	16	2.84	4.44	6.04
16	18	3.10	4.90	6.70
18	20	3.36	5.36	7.36
20	22	3.62	5.82	8.02
22	24	3.88	6.28	8.68
24	26	4.14	6.74	9.34
26	28	4.40	7.20	10.00
28	30	4.66	7.66	10.66
30	32	4.92	8.12	11.32
2 lbs	2.5	5.96	9.96	13.96
2.5	3.0	7.00	11.80	16.60
3.0	3.5	8.04	13.64	19.24
3.5	4.0	9.08	15.48	21.88
Each additional ½ pound over 4 pounds		add: 1.04	1.84	2.64

CANADA SHIPPING RATES

Up to:	
1	.20
2 oz	.37
3 oz	.54
4 oz	.71
5 oz	.88
6 oz	1.05
7 oz	1.22
8 oz	1.39
9 oz	1.56
10 oz	1.73
11 oz	1.90
12 oz	2.07
1 lb	2.58
2 lbs	3.10
3 lbs	4.10
4 lbs	5.10
5 lbs	6.10
6 lbs	7.10
7 lbs	8.10
8 lbs	9.10
9 lbs	10.10
10 lbs	11.10

For weights over 10 lbs, add $1.00 per pound.

Sorry, No Foreign Hardware Orders

A.P.P.L.E.™ PRODUCTS

AVAILABLE TO A.P.P.L.E. MEMBERS ONLY

SOFTWARE

☐	SAMP	AMPERMANAGER	$22.50
☐	SAPF	AP FILE	$18.50
☐	SFLR	AP GRAPHICS "The Filler"	$17.00
☐	SAM3	AP MAIL II (Version 3.0)	$30.00
☐	SAPT	AP TEST	$18.50
☐	SCAR	APPLESOFT CARPENTER //	$18.50
☐	SMLC	BIG MAC LC	$28.50
☐	SDISRE	DISK RECOVERY SYSTEM By Sof/Sys	$36.00
☐	SDAN	DRIVE ANALYZER DISKETTE	$55.00
☐	SEAP	E Z PILOT	$28.50
☐	SDIC	DICE	$12.50
☐	SHFI	HIGHER FONTS 1	$13.00
☐	SHTP	HIGHER TEXT PLUS	$27.50
☐	SMUL	MULTI-TRIEVE	$50.00
☐	SPGE	PROGRAM GLOBAL EDITOR (Version 2.0)	$18.50
☐	SPLE	PROGRAM LINE EDITOR (Version 2.0)	$15.00
☐	SS70	SOFT SEVENTY	$20.00
☐	SSYM	SYMBOL SIMON	$20.00
☐	STSS	THE SPREADSHEET	$75.00
☐	STSC	THE SPREADSHEET 1.0 to 2.0 converter	$5.00
☐	STUR	TURTLESOFT	$22.50

APPLE /// PRODUCTS

☐	SWW3	WORD WEAVER ///	$75.00
☐	HDR3	APPLE /// DISK DRIVE	$285.00

HARDWARE

☐	HSRMON	AMBER SCREEN MONITOR By Leading Edge	$95.00*
☐	HAPD	AP DRIVE	$235.00
☐	HAPC	AP CONTROLLER	$85.00
☐	HAKK	AP COOL KIT	$20.00
☐	HLC1	AP L CASE 1	$30.00
☐	HLC2	AP L CASE 2	$25.00
☐	H80C	APPLE //e 80 COLUMN CARD	$120.00
☐	HAUD	APPLESURANCE by Prometheus	$85.00
☐	HP16	AP RAM 16 K by Prometheuss	$50.00
☐	HP28	AP RAM 128 K by Prometheus DICE	$295.00
☐	HLC1A	w/AP L CASE 1	$37.50
☐	HLC2B	w/AP L CASE 2	$35.00
☐	HMPC	DUMPLING GX	$130.00
☐	HMPD	DUMPLING 64	$180.00
☐	HGRA	GRAPHITTI by Prometheus Please specify printer type: Epson (Graphtrax), C.ITOH (Pro- writer I, NEC), Okidata (Microline, Okiplot), IDS (460, 560, Prism), Apple Dot Matrix.	$85.00
☐	HPI1	HOBBY PROTOTYPING BOARD 11	$21.00
☐	HHOC	HOME CONTROLLER by Prometheus	$40.00
☐	HPAD	KOALA PAD TOUCH TABLET	$95.00
☐	HJOYST	MACH II JOYSTICK By Hayes	$34.00
☐	HJMACH	MACH III JOYSTICK By Hayes	$44.00
☐	HJOMA3	MACH III JOYSTICK By Hayes (for //e)	$49.00
☐	HMOB	MOCKINGBOARD by Sweet Microsystem	$159.00
☐	HMOD	MODEM II By Multi-Tech	$195.00
☐	MPANPR	PANASONIC KX1090 PRINTER	$325.00
☐	HSER	RS232 SERIAL INTERFACE CARD	$85.00
☐	HPANRI	PANASONIC P1090 RIBBONS	$14.00
☐	HPADDL	PADDLE-AD APPLE By SC RG	$25.00
☐	HMIL	PASCAL SPEED UP SYSTEM "The Mill" by Stellation II	$200.00
☐	HPRT	PROWRITER PRINTER	$380.00
☐	HPRC	Prowriter Ribbon cartridges	$5.00
☐	HRPT	REPEATERRRR	$20.00
		SOLID STATE DISK By Synetix	
☐	HSSS	147K SINGLE DRIVE	$275.00
☐	HSSU	147K UPGRADE KIT	$200.00
☐	HSSD	294K DUAL DRIVE	$500.00
☐	SCPM	SSD CPM Disk	$30.00
☐	SPAS	SSD Pascal Disk	$30.00
☐	SDOS	SSD DOS Disk	$30.00
☐	HFAN	SUPER FAN II Zener Ray	$79.00*
☐	HTPT	TALLY MT 160L PRINTER	$680.00
☐	HTPW	TALLY MT 180L PRINTER	$895.00
☐	HTRC	TALLY MT 160L RIBBON	$12.50
☐	HTRI	TALLY MT 180L RIBBON	$17.00
☐	HULTRA	THE ULTRATERM By Videx	$310.00
☐	HSUPMO	SUP R MOD By M S R Enterprises	$60.00
☐	HURB	ULTRA ROM BOARD/EDITOR (APB-102)	$149.00
☐	HVRC	VERSACARD by Prometheus(no cable)	$135.00
		Cables with VERSACARD	$15.00
		Cables only (specify)	$20.00
☐	HVPC	Parallel Printer Cable (6 ' long)	$20.00
☐	HVSL	Serial Cable (6 ' long)	$20.00
☐	HVSS	Serial Cable (1 ' long)	$20.00
☐	HVDR	V 1200 6 MB DISK DRIVE by Vista	$1295.00
☐	HCAR	Extra cartridges	$65.00
☐	HVIE80	VIEWMASTER 80 COLUMN CARD By Applied Engineering	$140.00

A.P.P.L.E. UTILITY DISK PAK

☐	SD6B	DISKPAK 6B-16	$18.50

A.P.P.L.E. GAMES

☐	SKO	KOALA GRAMS SPELLING 1	$31.50
☐	SSPI	SPIDER EATER	$24.00
☐	SCCO	COSMIC COMBAT	$12.50
☐	SD7	DISKPAK 7	$17.50
☐	SMCU	MUMMY'S CURSE	$12.50
☐	STHB	THUNDERBOMBS	$22.50
☐	SMIC	MICROBE	$30.00

A.P.P.L.E. LIBRARY

☐	PAUG	APPLE II USERS GUIDE	$13.00
☐	PBAD	BENEATH APPLE DOS	$17.00
		Call-A.P.P.L.E IN DEPTH	
☐	PID1	VOL I ALL ABOUT APPLESOFT	$7.00
☐	SIN1	VOl I Diskette	$16.50
☐	PID2	VOL II ALL ABOUT PASCAL	$11.00
☐	SIN2	VOL II Diskettes	$20.00
☐	PID3	VOL III ALL ABOUT DOS	$13.50
☐	SIN3	VOL III Diskettes	$15.00
	PCAB	Call-A.P.P.L.E. BACK ISSUES	
☐		1983 Back Issues(Jan to present)(each)	$2.50
☐		1982 Back Issues(each)	$2.50
☐		1981 Back Issues(each)(Jan, M/A, Jun, J/A, Sep, Oct, N/D qty limited)	$2.50
☐		1980 Back Issues(each) (M/A, May, Jun, N/D) qty limited	$2.00 each
		CALL —A.P.P.L.E. DISK OF THE MONTH	
☐	SCA6	6 mos. (start 1/84)	$55.00
☐	SCA12	12 mos. (start 1/84)	$95.00
☐	PPK4	BOUND SET (Call-A.P.P.L.E.) 1981	$18.50
☐	PPK5	BOUND SET (Call-A.P.P.L.E.) 1982	$18.50
		PEEKing at Call-A.P.P.L.E.	
☐	PPK2	VOL II 1979 Compendium	$10.00
☐	PPK3	VOL III 1980 Compendium	$15.00
		ANTHOLOGY VOLUMES (each)	$12.50
☐	SA1	VOL 1 (Jan thru Jun 1979)	
☐	SA2	VOL 2 (Jul thru Dec 1979)	
☐	SA3	VOL 3 (Jan thru Jun 1980)	
☐	SA4	VOL 4 (Jul thru Dec 1980)	
☐	SA5	VOL 5 (Jan thru Jun 1981)	
☐	SA6	VOL 6 (Jul thru Dec 1981)	
☐	SA7	VOL 7 (Jan thru Jun 1982)	
☐	SA8	VOL 8 (Jul thru Dec 1982)	
☐	SA9A	VOL 9a (Jan to March 1983)	
☐	SA9B	VOL 9b (March to June 1983)	
☐	SA10A	VOL 10a (July-Sep 1983)	
☐	SA10B	VOL 10b (Oct-Dec 1983)	
☐	SPA1	Pascal '81	
☐	PACD	THE APPLE II CIRCUIT DESCRIPTION	$18.50
		THE POWER OF VISICALC	
☐	PVIS	VOL I	$8.50
☐	PVI2	VOL II	$8.50
☐	PU65	USING 6502 ASSEMBLY LANGUAGE	$17.00
☐	PASL	ASSEMBLY LINES: THE BOOK	$17.00
☐		WHATS WHERE IN THE APPLE	
☐	PWWA	COMPLETE VOLUME	$17.00
☐	PWW2	THE GUIDE	$8.00
☐	PKID	KIDS AND THE APPLE	$16.00
☐	SPDS	PUBLIC DOMAIN SOFTWARE(each disk volume)	$4.00
☐	PELA	THE ELEMENTARY APPLE	$11.00
☐	PWTA	WOZ TALKS (VHS video tape)	$27.50

A.P.P.L.E. SERVICES AND ACCESSORIES

☐	HPAG	TYPING AID	$13.00
☐	HPIN	JEWELRY	$3.00
☐	FSOU	THE SOURCE	$60.00
☐	HFLS	FLIPSORT	$22.50
☐	HDBA	DISK BANKS BLANK DISKETTES	$4.50
☐	HVEB	Verbatim Datalife(Box of 10)	$25.00
☐	HCDB	Control Data Corp(Box of 10)	$22.00
☐	HGEN	BULK BLANK PACK (no jackets, set of 100)	$160.00
☐	FAPP	A.P.P.L.E. MEMBERSHIP ($25.00 first time application fee, $26.00 first years dues)	$51.00
☐	FREN	A.P.P.L.E. MEMBERSHIP RENEWAL	$26.00
☐	FLAT	LATE RENEWAL FEE	$5.00

Call -A.P.P.L.E. SPECIALS

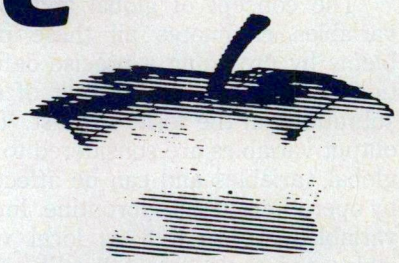
☐	4DISRE	DISK RECOVERY SYSTEM (Expires 2/28/84)	$32.50
☐	SPZAP	PROZAP (expires 3/31/84)	$22.50

Use Order Form on Reverse

* Indicates Introductory Price

FROM THE VERY CORE OF APPLE:

Earl of Varspace



Cornelis Bongers/Call —A.P.P.L.E. staff writer

Introduction

THIS article outlines how to implement and use global and local variables in Applesoft. A machine language routine that extends Applesoft with global and local variables is presented and details of operation are discussed extensively.

The prime reason I decided to develop a utility (named EARL: Extend Applesoft with Recursion and Local variables) to handle global and local variables is twofold. First, I needed it in order to (re)organize my BASIC subroutine library, which was becoming a big mess and second, I was, already for some time, under strong attack from several people who favor Pascal. Maybe it is sensible to spend a few more words on the latter. Generally speaking, most people who use Pascal seem to have a kind of contempt for their fellow computer users who work in BASIC. One of their main arguments against BASIC is that it is "impossible to write well structured programs in that language." When you ask what is meant by the latter, the same story is told over and over again; although BASIC supports subroutines, these subroutines are of not much value because variables and arrays used in a subroutine will influence variables and arrays with the same name in the main program. Next, they are quick to point out that in Applesoft, only the first two characters of a name are significant, so in large programs the probability of choosing the "wrong" (i.e. an already occurring) name nears one. Furthermore, they add, building a subroutine library of BASIC routines is impossible in Applesoft, because name conflicts will certainly arise sooner or later. Then they lean back in their chair, ready for the kill, and ask with a friendly smile how to handle recursion in BASIC.

As far as Applesoft is concerned, this article deals (as I hope) effectively with most of these arguments.

2. Local and Global Variables

For those not familiar with the concept of local and global variables, I will give in this section a brief introduction (directed towards the Applesoft implementation). The primary use of local and global variables is in subroutines. The idea is that it would be convenient if subroutines could be viewed as a black box with two holes; you put something in one hole and at the other hole something comes out. Only the input and the output of the black box (subroutine) are relevant and what happens inside should not influence things (variables) in the outside world (the main program).

As a simple example, suppose you want to calculate the sum of the elements of a one dimensional array. If this happens more than once in your program, it is efficient, in terms of memory space, to use a subroutine. The subroutine for the calculation of the sum looks as follows:

```
10   REM MAIN PROGRAM
     ........
100  GOSUB 1000
110  PRINT COUNT
200  END
     ........
     ........
1000 REM CALCULATE THE SUM
     OF AN ARRAY
1010 COUNT = 0
1020 FOR I=1 TO N
1030 COUNT=COUNT+A(I)
1040 NEXT I
1050 RETURN
```

As can be seen, it is assumed here that the array has name A and dimension N. The calculated sum is stored in the variable COUNT and the value of COUNT will further be processed by the main program.

When viewing the subroutine as a black box, the input consists of the array A and its dimension N, whereas the output consists of (the value of) COUNT. Note, however, that the variable COUNT is not the only output; the variable I can also considered to be an output variable, for the value of I is changed in the subroutine. This fact must be taken into account when the subroutine is invoked. If at that time a variable I in the main program has a certain value, this value will be changed after execution of the subroutine. In this simple example, the danger of conflicting names is remote, but in larger applications the danger is very real. When a subroutine uses, say 10 variables, it becomes already a tiresome task to check if none of the variables active in the main program has a name that starts with the same two characters as one of the variables in the subroutine. It goes without saying that matters get worse if you have several subroutines. By using a variable cross reference utility, it is of course always possible to check a program on conflicting names. However, usually it takes some time to print the cross reference and scanning a cross reference table is not a particularly pleasant job.

Matters get real bad if you want to use a BASIC subroutine from someone else (or a magazine) in your own program. This involves generally an extensive variable rename job and when you make a single error, the debugging process is cumbersome, as I know from my own experience.

EARL OF VARSPACE

The concept of global and local variables eliminates all these problems by providing a precise definition of the input and output of the subroutine in the form of a list. The output variables are considered to be global variables and can be affected by operations in the subroutine. Input variables are treated as local variables and their "main level" value cannot be changed by assignments to these variables in the subroutine. Furthermore, all variables and arrays not mentioned in the input/output list are automatically declared to be local (in the subroutine) and do not influence variables and arrays with the same name in the main program (or in the calling subroutine). In the example above, the array A and its dimension N would be defined as input variables and the variable COUNT as an output variable. The variable I would not be mentioned at all in the input/output list and would therefore not affect a variable named I in the main program.

The input/output list is a list of variables and arrays that are passed to the subroutine when it is invoked. When working with the EARL utility, the list must be specified behind the GOSUB lineno instruction (for precise details see the next section). For example, the program:

```
10   I=0:X=0:Y=0:& GOSUB
     100,X,Y
20   PRINT I,X,Y
30   END
     ....
     ....
100  X=1:Y=1:I=1:& RETURN
```

outputs 0,1 and 1, respectively. Note that I doesn't have the value 1 when control returns to the main program, although I is set to 1 in the subroutine. The reason is that I is not mentioned in the variables list behind the GOSUB statement. Consequently, upon return of the subroutine, I is set back to the value it had in the main program.

Now we go one step further. A severe limitation that still exists is that — although we are able to construct black boxes by making use of local variables — there is still a strong dependency between the subroutine and the main program (or other subroutines). This is that the names of the variables and arrays that are mentioned in the input/output list must match the names used in the subroutine. In particular, when passing arrays this constituted a problem. For example, suppose you have a BASIC subroutine that sorts the array A. Unfortunately, it turns out that the array you want to sort is named B, so before B can be sorted, it must be copied to A. Similar problems arise if you have a matrix multiplication or matrix inverse routine and in many other instances. It would therefore be very convenient if names of variables and arrays could temporarily be changed when they are passed to a subroutine.

You will be happy to hear (read) that EARL also supports this feature. The temporary name change is accomplished by specifying (optionally) a second parameter list. In the first list are mentioned the names of the main program variables and arrays you want to pass. The second parameter list contains the names the variables and arrays have in the subroutine. The program segment below illustrates the concept.

```
10   I=1:X=1:Y=1:& GOSUB
     100,X,Y
20   PRINT A,C,X,I
30   END
     ....
     ....
100  & > LINK, A,B
110  PRINT A,C,X,I
120  A=2:C=1:X=9
130  PRINT A,C,X,I
140  & RETURN
```

The first list is still specified behind the GOSUB instruction, but the second list is specified at the first line of the subroutine behind the keyword & > LINK. As can be seen, there is a one to one correspondence between the two lists here; the variable X in the main program becomes the variable A in the subroutine and Y becomes B.

The printout of the program above looks as follows:

```
1 0 0 0 (this results from the first
        PRINT in the subroutine)
2 1 9 0 (this results from the second
        PRINT in the subroutine)
0 0 2 1 (this results from the PRINT
        in the main program)
```

Let us next take a closer look at the values each variable gets when this program is executed. The first variable is I. Since I is not passed to the subroutine, it will have the value 0 in the subroutine. The output of the two PRINT statements in the subroutine shows that this is correct. The variable I is set back to 1 when control returns to the main program (see third line of the printout). The variable C falls in the same class as I. Although the value 1 is assigned to C in the subroutine, C is considered to be a new variable when control is back in the main program, so it gets the value 0. The variable X is passed to the subroutine (see line 10) and linked to the variable A. Thus the value of X in the main program (i.e. 1) is assigned to the variable A in the subroutine. As a result of the A=2 assignment at line 120, X assumes the value 2 when control returns to the main program (see third line of the printout). Note that there is also an assignment X=9 at line 120. This assignment doesn't influence the variable X in the main program, because the "main program variable" X has the name A in the subroutine. The variable X in the subroutine is thus considered being a local variable and will not be passed back to the main program.

The way of passing variables as described above is called "passing by reference." As opposed to passing by reference we have "passing by value." This method of passing is used if you want to pass only the value of a variable or array. Passing a variable or array by value implies that the original value of the variable or array cannot be clobbered in the subroutine. This is a necessity when you want to develop recursive subroutines (i.e. a subroutine that calls itself a number of times). Recursive subroutines are discussed extensively in section 5.

3. Syntax of the Earl Statements

When you want to use local and global variables in a subroutine, the first statement executed (in the main program) must be the statement:

```
& RETURN 0
```

Omission of this statement may lead to OUT OF MEMORY errors after a while (see section 7 for further explanations).

When a subroutine is invoked, a parameter list of variables and arrays must be specified behind the lineno of the GOSUB statement. In addition, an & symbol must be specified in front of the GOSUB keyword. An & GOSUB statement looks as follows:

```
& GOSUB lineno {,varex)}[/array}[ , array]]
```

where

```
varex : varname or (expression)
array : arrayname or (arrayname)
```

The & symbol links the EARL utility to Applesoft. When Applesoft encounters an &, control is transferred to location $3F5. At this location a JUMP instruction to EARL is installed during initialization (see section 9). EARL subsequently analyses and processes the & GOSUB statement. All variables and arrays specified in the (first) parameter list must already exist. For variables this means that a value must have been assigned to them (i.e. A=1) and for arrays that they must have been dimensioned. If this condition is not met, you will receive an OUT OF DATA error.

The counterpart of the & GOSUB statement is the & RETURN. When an & RETURN statement is executed, the program branches to the statement immediately following the most recently executed & GOSUB statement.

The second parameter list is optional, but when specified it must begin at the line with the line number specified in the & GOSUB statement. The second parameter list must be preceded by the keyword & > LINK. Thus,

```
& > LINK, parameter list
```

The parameters list has the same format as the parameter list behind the & GOSUB statement.

As can be deduced from the & GOSUB syntax, four different types of parameters can be passed. Distinguished by type of passing, these are:

A. Passing by Reference (Name)

VARIABLE — If the name of a variable is specified, this variable will be passed to the subroutine. Eventually, the name of the variable can temporarily be changed in the subroutine, by specifying a different name in the second parameter list. If a set to another value in the subroutine, the corresponding variable in the main program will have this value upon return. The passed variable may be a real, integer or string variable (thus no array element). In case a second parameter list is specified, the matching variable (name) must be of the same type as the variable being passed. If not, you will get a TYPE MISMATCH error.

ARRAY — Passing by reference of an array works similar to passing simple variables. In order to distinguish variables from arrays in the parameter list, a slash, rather than a comma, must be specified between the last variable and the first array. This applies to both parameter lists.

B. Passing by Value

(EXPRESSION) — If an expression, which must be enclosed by parentheses, is specified in the first parameter list, passing by value will be applied. A condition for a successful execution of the & GOSUB statement is that the second parameter list must contain a variable to which the value of the (evaluated) expression can be passed. If the expression is of type string, this variable must be a string variable. If the expression evaluates to a numeric result, the variable must be of type integer or type real.

The effect of passing by value is that the variable mentioned in the second parameter list assumes the value of the expression. This variable is further treated as a local variable, as is shown by the example below.

```
10   A=5:B=6:C=7
20   & GOSUB 100, (A),(10+5),
     C,("MARJO")
30   PRINT A,B,C
40   END
     ....
     ....
100  & > LINK, A,B,Q,A$
110  PRINT A,B,Q,A$
120  A=0:B=0:Q=0:& RETURN
```

The PRINT statement at line 110 will output the values 5, 15, 7 and the text MARJO, being the "values" passed to A, B, Q and A$. At line 120 A, B, and Q are set to zero, but this will only influence the value of C in the main program, since A was passed by value and B was not passed at all. The PRINT statement at line 30 will therefore output 5, 6 and 0, respectively.

Passing by value can be used to pass input values to subroutines. In this way, you can be sure that operations on these values will never affect the corresponding variables in the main program.

(ARRAY NAME) — Arrays can also be passed by value by enclosing the arrayname in parentheses. Passing by value of array expressions is not supported.

Next, some additional remarks regarding the second parameter list. The first is that the second parameter list may not have more parameters than the first. Violation of this condition leads to an OVERFLOW error. On the other hand, under certain conditions, the second parameter list may have less parameters than the first. For example, the construction:

```
10   X=1:Y=1:& GOSUB 100,X,Y
20   END
     ....
     ....
100  & > LINK,A
110  PRINT A,X,Y:& RETURN
```

is allowed. These statements have the effect that the variable X is named A in the subroutine. The variable Y, not mentioned in the second list, keeps its own name. The output at line 110 consists therefore of the values 1,0 and 1.

The following restrictions apply to the specification of a partial second parameter list.

1) If the first parameter list contains parameters that are passed by value, a corresponding variable or array must be specified in the second parameter list, otherwise you get an OUT OF DATA error.
2) If arrays are being passed and you want to change the names of the arrays in the subroutine, all variables that are passed must be specified. Thus the construction:

—◉→

```
10   A=1:B=1:DIM C(10)
20   & GOSUB 100,A,B/C
     ....
     ....
100  & > LINK,P/Q
```

is not allowed and leads to a SYNTAX error. Stated otherwise, if the second parameter list contains a slash, both lists must have the same number of variables before the slash.

It is allowable to pass arrays only. In that case, a slash must be specified immediately behind the line number of the & GOSUB statement and, if a second parameter list is specified, behind the keyword & > LINK. For example:

```
10   DIM A(10):& GOSUB 100/A
20   PRINT A(5)
30   END
     ....
     ....
100  & > LINK/B
110  B(5)=1:& RETURN
```

Finally, EARL supports also the equivalent of the Applesoft POP statement. Specification of

```
& POP
```

has the effect that control returns to the previous subroutine level. & POP works similar to & RETURN, except for the fact that when & POP is specified, execution continues at the statement following the & POP statement.

4. Examples

4.1. Matrix Multiplication

Useful applications for EARL are not hard to find. As a first example, I have rewritten my old matrix multiplication program (see *In The Heart of Applesoft, Micro,* February 1981). Listing 1 shows the result.

At line 30, three matrices are dimensioned and two of them are subsequently filled with random numbers. This is done with the subroutine at line 1000. Note that at line 40, K and L are passed by value, since these constitute input to the subroutine, whereas the array A is passed by reference. Next the matrices are printed by making use of the print subroutine at line 2000. Finally, the multiplication routine is invoked (see line 120). All variables and arrays are passed by value, except the matrix C into which the result of the multiplication is stored. Note that the loop variable K used at line 3030, does not influence the value of the first dimension of C, which is (in the main program) also assigned to the variable K.

```
10  REM
MATRIX MULTIPLICATION

15  &  RETURN 0: REM
INITIALIZE STACK
20  LET K = 2:L = 3:M = 4: REM
DIMENSIONS
30  DIM A(K,L),B(L,M),C(K,M)
35  REM
FILL MATRICES WITH
RANDOM INTEGERS
40  &  GOSUB 1000,(K),(L) / A
50  &  GOSUB 1000,(L),(M) / B
70  REM
PRINT MATRICES
80  &  GOSUB 2000,(K),(L),("A") /
    (A)
90  &  GOSUB 2000,(L),(M),("B") /
    (B)
110 REM
NOW MULTIPLY A AND B
120 &  GOSUB 3000,(K),(L),(M) /
    (A),(B),C
130 REM
PRINT RESULT
140 PRINT : PRINT "THE RESULT OF
    A * B IS"
145 &  GOSUB 2000,(K),(M),("C") /
    (C)
150 END
999 REM
SUBROUTINE TO FILL MATRIX
WITH RANDOM INTEGERS

1000  &  ) LINK,N,M / A
1010  FOR I = 1 TO N: FOR J = 1 TO
    M
1020  LET A(I,J) =  INT (10 *  RND
    (1))
1030  NEXT J,I
1040  &  RETURN
1999  REM
SUBROUTINE TO PRINT A MATRIX

2000  &  ) LINK,N,M,A$ / A:TB = 4
    0 / M
2005  PRINT : PRINT "MATRIX ";A$;
    " :": PRINT
2010  FOR I = 1 TO N: FOR J = 1 TO
    M
2020  PRINT A(I,J);: IF J <  ) M THEN
      PRINT  TAB( J * TB);
2030  NEXT J: PRINT : NEXT I
2040  &  RETURN
2999  REM
SUBROUTINE TO MULTIPLY
TWO MATRICES

3000  &  ) LINK,N,M,O / X,Y,Z
3010  FOR I = 1 TO N: FOR J = 1 TO
    O
3020  LET SUM = 0
3030  FOR K = 1 TO M
3040  LET SUM = SUM + X(I,K) * Y(
    K,J)
3050  NEXT K
3060  LET Z(I,J) = SUM
3070  NEXT J,I
3080  &  RETURN

MATRIX A :

3            5            8
3            1            7

MATRIX B :

2            3            8            9
1            4            1            8
5            2            0            4

THE RESULT OF A * B IS

MATRIX C :

51           45           29           99
42           27           25           63
```

The attractive point is that once you have developed a subroutine such as the matrix multiplication routine, you need only to registrate which arguments it requires and its starting line number. It is not necessary to remember what (scratch) variables or arrays are clobbered, as the main program or calling subroutine is protected against all operations done in the subroutine.

Generally, it might be a good idea to add an extra variable to the parameter list, which serves as error indicator. This error variable may be set to zero (in the subroutine) if no error occurs, else it may be set to the appropriate error number. The advantages of this approach are that the main program keeps control over things and that incorporation of the subroutine in other programs become easier. (If every subroutine starts printing its own error messages, you will soon end up in a big mess.)

4.2. A Decimal to Hex Converter

The next example concerns a dec-hex converter (see listing 2). The main program prints the hex equivalents of a series of decimal numbers. The actual dec-hex converter starts at line 1000. It requires input of a decimal number and it outputs the hex string in HEX$, the error return code in ER% and a text indicating the type of error in ER$. Note that — as opposed to the previous example — a considerable number of variables are used in the subroutine. Once again, you need never again worry about the well-known "side-effects," since the subroutine can change only the contents of HEX$, ER$ and ER%.

5. Examples That Use Recursion

Now the real work starts. What EARL can do in the field of recursion is roughly the same as what can be done in all the high-level languages. There are virtually no limitations, save the size of your memory. The 6502 system stack, which would limit the number of recursion levels severely, is simply abandoned (see section 7 for more details).

5.1. The Towers of Hanoi

The towers of Hanoi problem is one of the most elegant examples of how recursion can be used to solve a problem. The towers of Hanoi problem was recently discussed by Mr. K.M. Rudeen in *Creative Computing* (Nov 1982) and I borrowed some parts of his excellent problem description. The problem is described as follows.

There are 3 rods and N disks of increasing diameter. The disks can be stacked on any of the rods, forming towers. Suppose the disks are stacked in increasing size (from top to bottom) on one of the rods. We will call this rod the SOURCE rod. The task is now to move the N disks from the SOURCE rod to the DESTINATION rod, which may be any other of the two remaining rods. The moves must meet the following requirements:

1) In each step, only one disk can be moved from one rod to another rod.
2) A disk may never be placed over a smaller disk.
3) The remaining rod (SPARE rod) may be used to store disks temporarily.

```
]
10  REM
PRINT LIST OF HEX VALUES

15  &  RETURN 0: REM
INITIALIZE STACK
20  REM
PRINT DEC-HEX TABLE
30  TEXT : HOME : GOSUB 2000: REM

INIT PARMS
35  PRINT "DECIMAL          HEX"
39  REM
NOTE : ERROR IN LOOP BELOW

40  FOR I =  - 100 TO 1000 STEP 1
    00
50  &  GOSUB 1000.(I),HEX$,ER$,ER
    %
60  IF ER% THEN  &  GOSUB 3000,(E
    R%),(ER$): GOTO 80
70  PRINT I,HEX$
80  NEXT I
90  END
999  REM
DEC-HEX CONVERTER

1000  &  ) LINK,NUM,H$,E$,ER%
1010  LET ER% = 0: REM
INIT ERROR FLAG
1020  IF NUM )  = 2 ^ 16 THEN E$ =
    "VALUE TO LARGE":ER% = 1: GOTO
    1100
1030  IF NUM ( 0 THEN E$ = "VALUE
    LESS THAN ZERO":ER% = 1: GOTO
    1100
1040  LET R = 4096:H$ = ""
1050  FOR I = 1 TO 4:H =  INT (NU
    M / R)
1060  LET D$ =  STR$ (H): IF H )
    9 THEN D$ =  CHR$ (55 + H)
1070  LET H$ = H$ + D$:NUM = NUM -
    H * R:R = R / 16
1090  NEXT
1100  &  RETURN
1999  REM
INIT PARAMETERS

2000  LET HEX$ = "":ER$ = "":ER% =
    0: RETURN
2999  REM
ERROR HANDLER

3000  &  ) LINK,E%,E$
3005  PRINT  CHR$ (7);
3010  IF ER% = 99 THEN  PRINT "FA
    TAL ERROR": STOP
3020  PRINT : PRINT E$: PRINT
3030  &  RETURN
DECIMAL          HEX

VALUE LESS THAN ZERO

0                0000
100              0064
200              00C8
300              012C
400              0190
500              01F4
600              0258
700              02BC
800              0320
900              0384
1000             03E8
```

EARL OF VARSPACE

As Mr. Rudeen argues, in order to apply recursion to solve this problem, it must be broken up in a number of subproblems, which are either trivial or a simpler version of the original problem. The towers of Hanoi problem can be broken up into the following three subproblems:

1) If N>1 then move N-1 disks from the SOURCE rod to the SPARE rod.
2) Move one disk from the SOURCE rod to the DESTINATION rod.
3) If N>1 then move N-1 disks from the SPARE rod to the DESTINATION rod.

When numbering the rods from 1 to 3, the number of the SPARE rod equals 6 — number of SOURCE rod — number of DESTINATION rod. For example, if the SOURCE rod is rod 2 and the DESTINATION rod is rod 3, the SPARE rod is rod 6-2-3=1.

It is rather surprising that these observations are all that is needed to solve the problem. By making use of recursion, the three steps above can immediately be translated in a BASIC program. The program is displayed in listing 3 and the recursive subroutine at line 100 does the job. The subroutine is called at line 50 in the main program. The arguments of the call are: the number of disks that need be moved, the number of the source rod and the number of the destination rod. The statements at lines 110 and 130 in the subroutine pass a simpler version of the problem (i.e. the problem to move N-1 disks) to the same subroutine. However, as soon as the problem becomes trivial, which is the case if N=1, the required move that solves it is printed (see line 120).

Note that passing by value is applied here. This is essential for the values of SOURCE, SPARE, DESTINATION and N must not be changed by operations at lower level subroutines. If this were the case, (a.o.) the PRINT statement at line 120 would output the wrong values.

It is not that difficult to program the towers of Hanoi problem in normal BASIC. It is a fact however, that you must set up a stack simulation and administration first and this makes the programming job more cumbersome.

A question that is related to the towers of Hanoi problem is: How many moves are (minimally) required to get the job done. This number can be calculated in several ways. First, an extra variable, NO, can be passed to the subroutine (by reference). NO is then incremented each time a move is printed. When control returns to the main program, NO will contain the number of moves. Alternatively, we may derive a formula from the 3 subproblems mentioned above. If the number of moves required to move N disks from the SOURCE rod to the DESTINATION rod is designated by F(N), we have:

F(N) = F(N-1) + 1 + F(N-1) = 2 * F(N-1) + 1

and

F(0) = 0

```
10  REM
TOWERS OF HANOI

15  &  RETURN 0: REM
INITIALIZE STACK
20  INPUT "ENTER NUMBER OF DISKS
    ";N
30  INPUT "ENTER SOURCE AND DESTI
    NATION ROD ";SOURCE,DEST
40  PRINT : PRINT "STRATEGY:": PRINT

50  &  GOSUB 100,(N),(SOURCE),(DE
    ST)
60  END
90  REM
THE MOVE SUBROUTINE

100  &  ) LINK,N,SOURCE,DEST
105  LET SPARE = 6 - SOURCE - DES
    T
110  IF N ) 1 THEN  &  GOSUB 100,
    (N - 1),(SOURCE),(SPARE)
120  PRINT "MOVE DISK FROM ROD ";
    SOURCE;" TO ROD ";DEST
130  IF N ) 1 THEN  &  GOSUB 100,
    (N - 1),(SPARE),(DEST)
140  &  RETURN

ENTER NUMBER OF DISKS 4
ENTER SOURCE AND DESTINATION ROD 1,2

STRATEGY:

MOVE DISK FROM ROD 1 TO ROD 3
MOVE DISK FROM ROD 1 TO ROD 2
MOVE DISK FROM ROD 3 TO ROD 2
MOVE DISK FROM ROD 1 TO ROD 3
MOVE DISK FROM ROD 2 TO ROD 1
MOVE DISK FROM ROD 2 TO ROD 3
MOVE DISK FROM ROD 1 TO ROD 3
MOVE DISK FROM ROD 1 TO ROD 2
MOVE DISK FROM ROD 3 TO ROD 2
MOVE DISK FROM ROD 3 TO ROD 1
MOVE DISK FROM ROD 2 TO ROD 1
MOVE DISK FROM ROD 3 TO ROD 2
MOVE DISK FROM ROD 1 TO ROD 3
MOVE DISK FROM ROD 1 TO ROD 2
MOVE DISK FROM ROD 3 TO ROD 2
```

```
10  REM
CALCULATE NUMBER OF MOVES (HANOI)

15  &  RETURN 0: REM
INITITIALIZE STACK
20  INPUT "ENTER NUMBER OF DISKS
    ";N
25  LET NO = 0: REM
NO MUST BE INITIALIZED
30  &  GOSUB 100,(N),NO
40  PRINT : PRINT NO;" MOVES ARE
    NEEDED"
50  END
99  REM
MOVE COUNTER

100  &  ) LINK,N,NO
110  LET NO = 0: REM
RETURN 0 IF N=0
120  IF N THEN  &  GOSUB 100,(N -
    1),NO:NO = 2 * NO + 1
130  &  RETURN

ENTER NUMBER OF DISKS 15

32767 MOVES ARE NEEDED
```

To calculate F(N), we also may use recursion. The resulting program is displayed in listing 4. The subroutine at line 100 calculates F(N) for a given N and returns the value of F(N) in the variable NO. As easily can be checked, if N=0, the subroutine returns the value zero in NO. On the other hand, if N differs from zero, the subroutine calls itself (see line 120). The purpose of this call is to calculate F(N–1). Since the value of F(N–1) is also returned in NO, the statement NO=2*NO+1 must be executed to calculate F(N). Finally, the & RETURN statement at line 130 returns the value of F(N), stored in the variable NO, to the previous level.

Of course, those readers whose math is not yet undermined by years of computer usage, will prefer the third solution, which consists of solving the difference equation and leads to:

$$F(N) = 2 \wedge (N) - 1$$

5.2 Permuations

Another nice example of recursion is provided by the generation of permutations. The problem is to generate all permutations of the numbers 1 to N and print these in an "increasing" sequence. For N is 3, this should result in:

```
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

Before proceeding to read the solution, try to make a permutation generator yourself in "normal" BASIC. After that you will probably much better understand and appreciate the power of recursion.

Let's start by reserving a one dimensional array PER with dimension N. PER is initialized to the values 1 to N (see listing 5, lines 30 and 35). Next, we break up the permutation problem in N subproblems.

```
10  REM
PRINT N PERMUTATIONS

15  &  RETURN 0: REM
INITIALIZE STACK
20  INPUT "ENTER NUMBER OF ELEMEN
    TS ";N: PRINT
30  DIM PER(N):FIRST = 1
35  FOR I = 1 TO N:PER(I) = I: NEXT
    : REM
FILL ARRAY
40  &  GOSUB 100,(FIRST),(N) / (P
    ER)
50  END
99  REM
THE PERMUTATION GENERATOR

100  &  ) LINK,FIRST,N / PER: IF
     FIRST (  ) N GOTO 130
110  FOR I = 1 TO N: PRINT PER(I)
     ;" ";: NEXT : PRINT
120  &  RETURN
130  FOR I = FIRST TO N
140  LET TEMP = PER(FIRST):PER(FI
     RST) = PER(I):PER(I) = TEMP
150  &  GOSUB 100,(FIRST + 1),(N)
      / (PER)
160  NEXT
170  &  RETURN

ENTER NUMBER OF ELEMENTS 4

1  2  3  4
1  2  4  3
1  3  2  4
1  3  4  2
1  4  2  3
1  4  3  2
2  1  3  4
2  1  4  3
2  3  1  4
2  3  4  1
2  4  1  3
2  4  3  1
3  1  2  4
3  1  4  2
3  2  1  4
3  2  4  1
3  4  1  2
3  4  2  1
4  1  2  3
4  1  3  2
4  2  1  3
4  2  3  1
4  3  1  2
4  3  2  1
```

```
10  REM
SORT ARRAY OF N ELEMENTS

15  &  RETURN 0: REM
INITIALIZE STACK
20  INPUT "NUMBER OF ELEMENTS TO
     BE SORTED ? ";N
30  PRINT : DIM NUM(N)
40  FOR I = 1 TO N:NUM(I) =  RND
    (1): NEXT
50  &  GOSUB 100,(1),(N) / NUM: REM

SORT NUM
60  FOR I = 1 TO N: PRINT I,NUM(I
    ): NEXT
70  END
99  REM
THE QUICKSORT SUBROUTINE

100  &  ) LINK,FIRST,LAST / A
110  LET P1 = FIRST:P2 = LAST:MID
      = A((FIRST + LAST) / 2)
120  IF A(P1) ( MID THEN P1 = P1 +
     1: GOTO 120
130  IF A(P2) ) MID THEN P2 = P2 -
     1: GOTO 130
140  IF P1 ) P2 GOTO 170: REM
PASS COMPLETED
150  LET TEMP = A(P1):A(P1) = A(P
     2):A(P2) = TEMP
160  LET P1 = P1 + 1:P2 = P2 - 1:
      GOTO 120
170  IF FIRST ( P2 THEN  &  GOSUB
     100,(FIRST),(P2) / A
180  IF P1 ( LAST THEN  &  GOSUB
     100,(P1),(LAST) / A
190  &  RETURN

NUMBER OF ELEMENTS TO BE SORTED ? 15

1              .066365339
2              .187514164
3              .255693611
4              .444066158
5              .485920949
6              .505683995
7              .537647585
8              .603199523
9              .668783879
10             .839690955
11             .852083839
12             .866859601
13             .930325549
14             .962219281
15             .990148649
```

Suppose, we know how to generate all permutations of the array elements PER(2) to PER(N). Then we can solve the problem by applying the following strategy:

1) Generate all permutations of the array elements 2 to N
2) Exchange PER(1) and PER(2) and do step 1
3) Exchange PER(1) and PER(3) and do step 1
   .....
   .....
N) Exchange PER(1) and PER(N) and do step 1

Note that the first step can also be written as: exchange PER(1) and PER(1) and generate all permutations of the array elements 2 to N. Note further that for a successful execution it is required that after each step has been executed, the elements of the PER array must satisfy the condition PER(I)=I, since I is 1 to N, otherwise the exchange trick will not work. This indicates that we have to use passing by value in this situation.

The final question is when to print a permutation. As can be deduced from the structure of the strategy above, this must be done when control is at the deepest level and this level is reached when a "permutation" of one element must be generated.

The permutation generation routine starts at line 100. Lines 130 to 160 perform the N steps mentioned above. At line 150, the routine is called to solve a "simpler" subproblem, which is to generate all permutations of the numbers stored in the array elements PER(FIRST+1) to PER (N). Note that the "current" problem is to generate all permutations of the elements stored in PER, (FIRST) to PER(N). If FIRST=N, the problem becomes trivial since in that case there is only one permutation. At this point, the PER array is printed (see line 110).

The outlined method of generating permutations is not the most efficient in terms of execution time. However, it has the advantage that it is easy to understand and to program.

5.3. Quick Sort

The final example concerns a quick sort routine (see listing 6). At lines 30 to 50 an array is initialized with random numbers and sorted by means of a call to the quicksort subroutine at line 100. I copied this routine from *Algorithms + Data Structures = Programs* (p. 79) by N. Wirth. Although the routine was written in Pascal, conversion to BASIC was straightforward. To give the Pascal fanatics something to complain about, I added some GOTO statements, but by making use of my & IF-THEN-ELSE utility (see *Call A.P.P.L.E.*, Vol. V, Number 10), most GOTO's can easily be avoided.

Since an array is passed, you may run out of memory sooner than anticipated. With quicksort, the number of recursive subroutine calls needed to sort the array depends not only on the dimension of the array but also on its contents. In any case, if the array is filled with randoms, there will be no problems (unless you are extremely unlucky) as long as the dimension is less than 200.

6. Things You Need to Know

When working with EARL, there are a few things you should keep in mind. The first and most important is that the first statement in the program must be the & RETURN 0 statement. This statement has about the same function as the Applesoft RUN command. If you enter RUN, Applesoft clears the 6502 system stack and resets the BASIC pointers before executing the program. Similarly, & RETURN 0 clears the secondary stack maintained by EARL and initializes its stack pointer. Subsequently, a CLEAR command is issued to reset the BASIC pointers to their proper values. Hence it is very important that & RETURN 0 be the first statement executed, because all variables defined before & RETURN 0 is encountered will be cleared.

If you forget to specify the & RETURN 0 statement, you may eventually receive an OUT OF MEMORY error. However, this will only happen when the program is repeatedly interrupted (by [Ctrl-C] or errors) and it is restarted each time with RUN. The secondary stack will then grow larger and larger until it collides with the stringpool. If an (error-free) program runs without being interrupted, the secondary stack will be cleared automatically if the end of the program is reached.

If you want to reset HIMEM or LOMEM to another value, insert the appropriate statement directly behind the & RETURN 0 statement. Never attempt to change HIMEM or LOMEM from within a subroutine invoked with & GOSUB, as this may have disastrous results. Note that some utilities reset HIMEM, so don't install these utilities with a BRUN command from within a subroutine.

As to levels of nesting, you may nest & GOSUB statement as deep as you want, the only limitation being the amount of your free memory. The secondary stack is namely built up in the area starting behind the end of your program (or at LOMEM). Unlike Applesoft, EARL doesn't use the 6502 stack, so the well-known Applesoft limitation (maximally 24 levels of nesting) does not apply here.

A next important point concerns the passing of arrays. When an array is passed to a subroutine, it is actually copied to another position in memory. This means that if you work with large arrays, you may get an OUT OF MEMORY error even when passing a single array. As a general rule, when passing a single array, the free space available just before the subroutine is invoked must at least be:

(number of array elements) * K + 100

where

K = 5 when passing a real array
K = 3 when passing a string array
K = 2 when passing an integer array

To get some insight into the rate at which memory is consumed when an & GOSUB statement is executed, you can specify the statement: PRINT FRE(0) as the first statement in your subroutine. Especially when using recursive subroutines, this will give you a good indication as to how "deep" you can go.

Regarding normal GOSUB/RETURN statements, these can be used freely in a program containing & GOSUB/& RETURN statements. However, you cannot not use RETURN as a substitute for & RETURN or & RETURN as a substitute for RETURN. This implies that no GOSUB statement in a routine invoked with & GOSUB may be active at the time the & RETURN statement is encountered, else you will get a RETURN WITHOUT GOSUB error. This will, for example happen in the program below:

```
10   A=1:& GOSUB 100,A
     ....
     ....
100  & > LINK, B
110  GOSUB 1000
     ....
     ....
1000 B=10:& RETURN:REM A
     RETURN MUST BE
     EXECUTED FIRST
```

Summarizing, the type of the return statement (i.e. RETURN or & RETURN) must match the type of the associated GOSUB statement. Violation of this condition leads to a RETURN WITHOUT GOSUB error.

Regarding loops, the same rules apply as when using these in a "normal" Applesoft subroutine. Upon return of the subroutine all active loops in the subroutine will be killed. A subsequent NEXT in the main program (or the calling subroutine) will lead to a NEXT WITHOUT FOR error (unless of course loops are active in the main program).

An & GOSUB statement may be inserted in a (nested) loop, for example:

```
10   FOR I=1 TO 100
20   & GOSUB 100,(I)
30   NEXT
40   END
     ....
     ....
100  & > LINK,J: J=J+1
110  PRINT J: & RETURN
```

For the experts, something to think about: You don't need to worry about the total number of loops that are active on the various (higher) subroutine levels. Only the number of active loops on each subroutine level is relevant; this number must be smaller than 11 (the normal Applesoft restriction), otherwise you will receive an OUT OF MEMORY error.

7. How it Works

Once I had the idea worked out, implementation of EARL turned out to be a comparatively simple operation. Although the utility has considerable size, the coding is straightforward and should be easy to understand. Before explaining the internals of EARL, we will first briefly examine the organization of the Applesoft variables, arrays and strings. When you RUN a BASIC program and all your variables, arrays and strings have been initialized, memory looks as displayed in figure 1.

All BASIC variables have a length of 7 bytes and the first two bytes are reserved for the name of the variable. Arrays have no fixed length, but similar to variables, the first two bytes contain the name of the array.

Variables and arrays that are of type string contain a pointer(s) to their associated string(s). These strings reside in the stringpool, which is build up at the end of memory. The relevant BASIC pointers are:

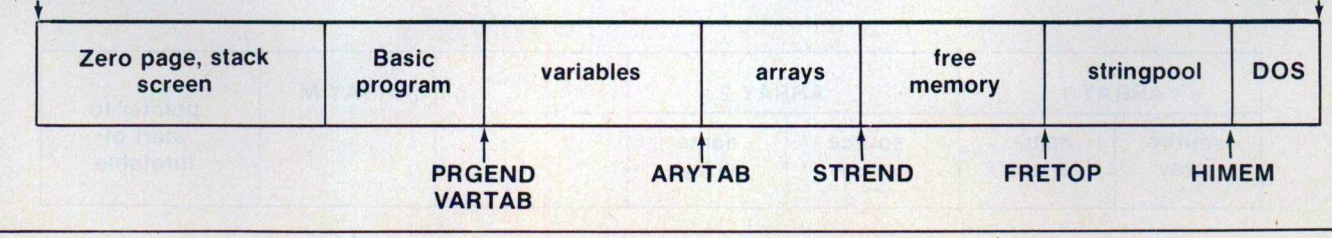
PRGEND	**Points to the end of the program**
VARTAB	**Points to the start of the variable space (LOMEM)**
ARYTAB	**Points to the start of the array space**
STREND	**Points to the end of the array space**
FRETOP	**Points to the bottom of the stringpool**
HIMEM	**Points to the top of the stringpool**

The main problem I encountered when I decided to implement local and global variables in Applesoft was: How can a distinction be made between variables that are local and those that are global. The solution I found makes use of the BASIC pointers to create an entire new variable/array space and stringpool. The variables and arrays that are mentioned in the & GOSUB parameter list are copied from the old variable/array space to the new one. If a second parameter list is specified (behind & > LINK), the names of the variables and arrays are adjusted after the move. So, after Applesoft executes an & GOSUB statement, memory looks as displayed in figure 2.

When an & RETURN statement is executed, all that need be done is to copy the variables/arrays that were passed by reference back to the old variable/array space and to reset the BASIC pointers to their old (previous) values. Note that the variables and arrays that were passed by value need not be copied back to the old variable/array space. Variables and arrays that were created in the subroutine also are not copied to the old variable/array space, so these cannot influence in any way variables/arrays with the same name on the previous level.

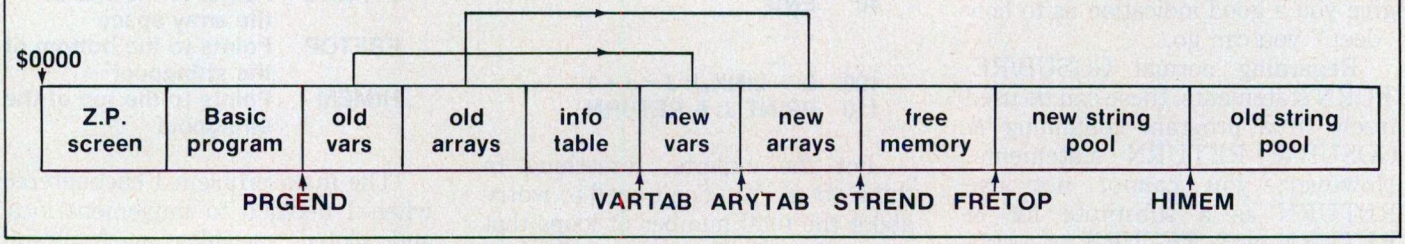
—◉→

Figure 1 Memory Map before execution of & GOSUB



EARL OF VARSPACE

Figure 2 Memory Map after execution of & GOSUB



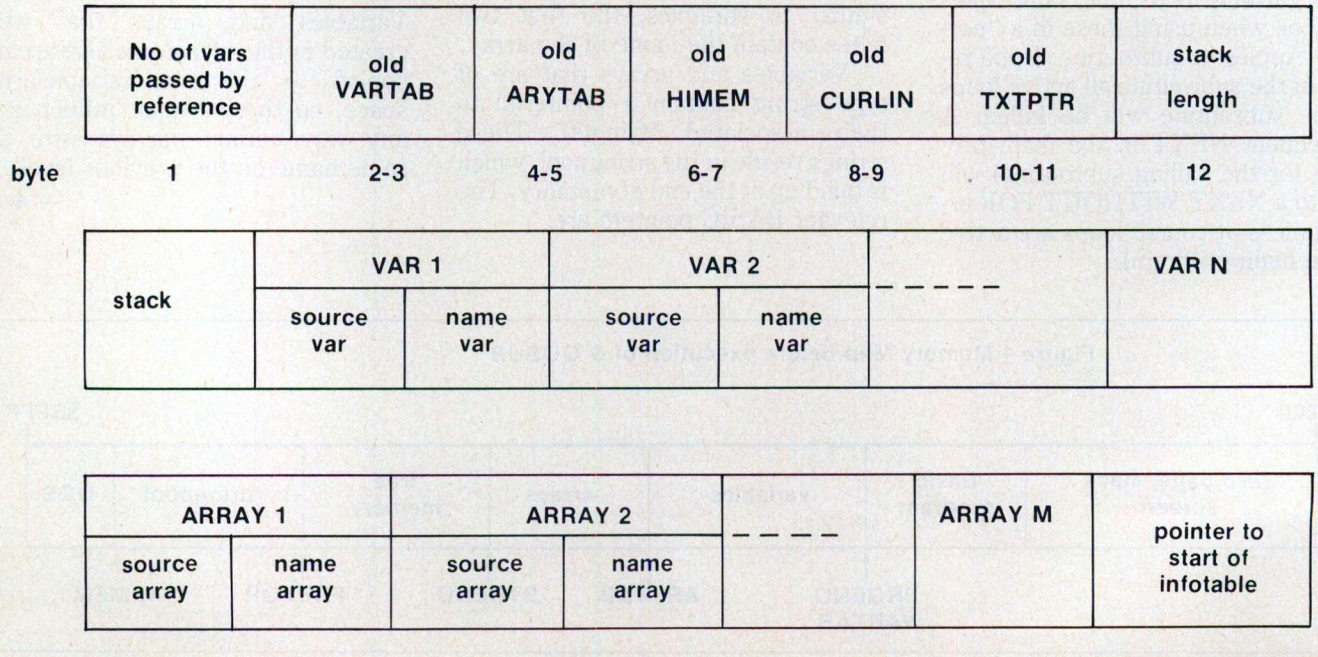
Of course, we need to keep track of the variables and arrays that were passed by reference to the new variable/array space, so that we can move them back when & RETURN is executed. For this purpose the following information is needed:

1) The name the passed variable/array has in the subroutine
2) The source address of the variable/array (in the old variable/array space)

If we know the name of the variable or the array, it is a simple operation to find it in the new variable/array space (we let Applesoft do the work). Thereafter it can be copied to the source address (except for the first 2 bytes which form the name).

The information regarding the passed variables and arrays is stored in an info table by the part of EARL that handles the & GOSUB statement. This table is built up at the start of free storage (behind the current value of STREND). The info table is displayed in figure 3. The first entry of the table contains the number of variables that is passed by reference. This information is needed in order "to know" — upon & RETURNing — when we have to start passing back arrays. Bytes 2–7 contain the old values of VARTAB, ARYTAB and HIMEM. The BASIC pointers are set back to these values if, and only if, the & RETURN statement is successfully executed. Bytes 8–9 and 10–11 contain the linenumber of the & GOSUB statement and the associated value of the textpointer. Note that Applesoft stores these values on the 6502 stack. To be honest, I did this also in the first version of EARL but soon discovered that this led — in particular when working with recursive subroutines — to stack overflow. By storing these pointers in "normal" memory, the stack is not charged, which implies that you can invoke the subroutine again and again without getting stack overflow. After I put CURLIN and the textpointer in the info table, I thought that all stack-related problems were solved. Unfortunately however, I overlooked the loops. Since the information regarding the active loops is also stored on the 6502 stack, it turned out that recursive subroutines that made use of loops (i.e. the permutation example) led to OUT OF MEMORY errors. Thus, I decided there was only one effective way to deal with the stack, which involves saving the active part of the stack in the info table. (If the stack contains no relevant information, nothing is saved). After the stack is

Figure 3 The organization of the info table



copied to the info table, the 6502 stackpointer is reset, to indicate the stack being empty. As a nice side effect, this gives EARL more control over stack related matters. For example, in the previous version of EARL you could specify:

```
10    FOR I=1 TO 10
20    & GOSUB 100,I
30    END
      ....
      ....
100   & > LINK,J:NEXT
```

without getting an error message. This is not longer so. Since the 6502 stack pointer indicates an empty stack, Applesoft will nicely print a NEXT WITHOUT FOR error.

After the stack, the storage area of the passed variables and arrays starts. As outlined above, each variable or array occupies four bytes in the infotable.

The last two bytes of the info table contain a pointer to the start of the info table (the length of the table is not fixed). Note that this pointer is identical to the previous value of STREND.

The general concept should be clear by now (if not, don't hesitate to read the previous part again, after all it took me some time before I had it all worked out and on paper). If an additional variable/array space can be created once, it can as well be created multiple times by applying the same technique. So each time an & GOSUB statement is executed, a new variable/array space will be created and on each & RETURN, the current variable/array space is designated as free memory after all variables and arrays mentioned in the infotable have been copied back to the previous variable/array space.

At this point there is still a nasty problem left. If an & RETURN statement is processed, EARL must determine if there is an info table just below the address referenced by (the current value of) VARTAB. If so, the & RETURN can be executed, but if not, a RETURN WITHOUT GOSUB error should be given. The problem now is: how do we know an info table is present. A related problem is that major problems arise if a program is interrupted in a subroutine, since a subsequent RUN statement doesn't clear the secondary stack (i.e. the stack of variable/array spaces), since Applesoft has no way of knowing that a secondary stack exists. Similarly, EARL has no way of knowing that a RUN command has been issued, thus EARL also does not know when the secondary stack need be cleared.

The only method I found to deal with these problems was the introduction of a separate statement (i.e. & RETURN 0) which clears the secondary stack. When executed, this statement restores HIMEM to its original value and copies the pointer to the end of the BASIC program in VARTAB.

In more detail, the following procedure is applied. Location 0–2 and 3–5 are initialized upon an Applesoft cold start with two JMP instructions. These point to the warm entry of Applesoft and an internal Applesoft routine to print a string. The warm start vector is used only by the old disk-based RAM Applesoft, and no internal routines have ever referenced STROUT, so I decided to reserve the associated memory locations for a secondary stack indicator. When an & GOSUB statement is executed, EARL determines if the original JMP instructions are present at 0–2 and 3–5. If so, BRK instructions are inserted at 0 and 3 and the main level values of VARTAB and HIMEM are stored at 1–2 and 4–5, respectively (The purpose of the BRK instructions is to halt programs that use the JMP's). If, on a subsequent execution of an & GOSUB instruction, EARL detects that the original JMP instructions are not present, no changes are made. In a sense, & RETURN performs the reverse operation. When EARL executes an & RETURN, a check is done for the presence of the original JMP instructions. If these are present, a RETURN WITHOUT GOSUB error is generated, since the secondary stack must then be empty. On the other hand, if the original JMP instructions are not present, the VARTAB retrieved from the info table is compared with the value of the "main-level" VARTAB stored at 1–2. If the pointers differ, no action is taken, but if they match, control must be back at the main level (after the & RETURN has been executed). To prohibit execution of subsequent & RETURN statements, the original Applesoft JMP instructions are then restored.

Under abnormal circumstances, when a program is error free and is not interrupted by control C, this method works perfectly. If all & GOSUB statements and & RETURN statements properly match, the secondary stack will be empty upon termination of the BASIC program and the original Applesoft JMP instructions will reside at 0–5. However, since programs will crash occasionally, after awhile you generally get the same results as when you specify:

```
10    GOSUB 20
20    REM
```

and enter GOTO 10 twenty-eight times.

8. In Case of an Error

If EARL reports an error, there are many possible causes. To give you some clue as to what might be wrong, the most frequent problems are listed in TABLE 1, indexed to error message.

Important Note: When EARL detects an error while executing an & GOSUB statement, the error message will usually refer to the line number of the line that contains the & GOSUB statement. However, the error may also be caused by a specification error in the second parameter list. So, if you get an error message, check both parameter lists.

Execution of the & RETURN 0 statement prevents unnecessary OUT OF MEMORY errors. The statement checks if locations 0–5 contain the original Applesoft JMP instructions. If so, no further actions are taken; only a CLEAR command is issued. In the other case, the pointer to the end of the BASIC program is copied in VARTAB and HIMEM is set to the values stored at locations 4–5. Next, the JMP instructions are restored and a CLEAR is done to reset all the other BASIC pointers to the right values.

Warning: it goes without saying that clobbering locations 0–5 may have disastrous results after an & GOSUB statement has been executed.

—●→

Table 1 Error Messages and Probable Causes

GOSUB ERROR

- This is the most serious error you can get. The error occurs if EARL detects that its secondary stack information is clobbered by another utility or by POKE statements. Unless you are able to reset the relevant pointers "by hand," the best thing you can do is to reboot your system.

SYNTAX ERROR

- No & symbol was specified in front of an & GOSUB or & RETURN 0 statement.
- No parameter list was specified behind the line number of the & GOSUB statement.
- One of the expressions passed between parentheses is syntactically incorrect. Note that string literals that are passed must have an opening *and* a closing quote (").
- The parameter list contains more than one slash.
- The number of variables before the slash in the first parameter list differs from the corresponding number in the second list.
- A line starting with & > LINK was encountered in the course of normal program execution and not via an & GOSUB statement.

OVERFLOW ERROR

- The second parameter list contains more parameters than the first.

TYPE MISMATCH ERROR

- On passing by reference: the type of the variable/array in the first parameter list differs from the type of the associated variable/array in the second parameter list (the types must be identical)
- On passing by value: the type of the variable/array in the first parameter list was numeric and the type of the corresponding variable/array in the second parameter list was a string, or vice–versa.

RETURN WITHOUT GOSUB ERROR

- No & symbol was specified in front of an & GOSUB or & RETURN statement.
- A RETURN statement was encountered but no MATCHING GOSUB statement had been executed.

OUT OF DATA ERROR

- A variable specified in the first parameter list was not created (referenced) yet.
- An array specified in the first parameters list was not dimensioned.
- Passing by value was indicated but no associated variable/array was specified in the second parameter list.

OUT OF MEMORY ERROR

- No & RETURN 0 statement was specified.
- The secondary stack collided with the stringpool. This is usually caused by too many levels of nesting.

REDIMENSIONED ARRAY ERROR

- The length of an array to be passed back to the previous variable/array space differed from the length of the original array. In principle, this error can only occur if a clear utility was used to erase the array and the array was subsequently dimensioned again, but with different dimensions.

FORMULA TOO COMPLEX ERROR

- This error should (hopefully) not occur. If it does, try to reduce the complexity of expressions passed. If this doesn't help, contact the author.

OTHER ERROR

- You may get all kind of weird results if the secondary stack occupies (a part of) a High-Res page and you execute an HGR or HGR2 command. This will usually erase (a part of) the secondary stack. Problems of this type can be avoided by specifying LOMEM:4*4096 (for HGR) or LOMEM:6*4096 (for HGR2) directly behind the & RETURN 0 statement.

If after working with EARL you have any useful suggestions to extend the error list, I would be happy to hear them.

9. Installing EARL

EARL is almost $500 bytes long and starts at $9000. It was assembled with Big Mac, but no use was made of any special Big Mac features, so you should have no trouble assembling the source with your own assembler. If you don't have an assembler, you have to type in the hex codes. This is a difficult job and few people manage to do it without errors, so, please double check before writing an angry letter that things don't work correctly.

After the object has been saved on disk, EARL can be installed by issuing a BRUN command. This has the effect that EARL's initialization routine is executed, which sets HIMEM and the pointer to the bottom of the stringpool to the start of EARL. If you install EARL from within an Applesoft program, insert the BRUN instruction as the first statement in your program, since any active strings will be overwritten by EARL.

10. Final Remarks

To my Knowledge EARL is bug free (The program I wrote to test EARL has a length of 11K). Still, there always remains the possibility that I overlooked something. Therefore, if you encounter anything that looks like a bug, I would appreciate very much if you would so advise me.

Finally, I wish you good luck working with EARL and hope it serves you well as your personal (VAR)SPACE controller.

Cornelis Bongers
Erasmus University
Postbox 1738
3000 DR Rotterdam
The Netherlands

Acknowledgment: I wish to thank Hans Geilenkirchen for his many valuable suggestions and comments during the development of EARL.

```
*9000.94FE

9000- A9 00 85 73 85 6F A9 90
9008- 85 74 85 70 A9 17 8D F6
9010- 03 A9 90 8D F7 03 60 C9
9018- B0 F0 11 A2 00 C9 B1 F0
9020- 08 CA C9 A1 F0 03 4C C9
9028- DE 4C A0 93 20 F2 91 A5
9030- 02 C9 D4 D0 11 A2 02 B5
9038- 68 95 00 B5 72 95 03 CA
9040- D0 F5 86 00 86 03 20 B1
9048- 00 20 0C DA A5 B9 85 18
9050- 48 A5 B8 85 17 48 20 41
9058- D9 20 D3 91 A0 00 84 07
9060- 84 09 84 FD 84 1F A0 04
9068- C8 B1 17 D9 BB 91 D0 0E
9070- C0 0A D0 F4 38 98 65 17
9078- 85 17 90 02 E6 18 A2 03
9080- 86 1E 20 B7 00 F0 39 D0
9088- 27 C9 28 D0 21 E6 1F 20
9090- B1 00 F0 2C C9 22 D0 09
9098- 20 B1 00 F0 23 C9 22 D0
90A0- F7 C9 28 F0 E8 C9 29 D0
90A8- E6 C6 1F D0 E2 2C E6 1E
90B0- AA 20 B1 00 F0 0D E0 2C
90B8- F0 CF E0 CB F0 CB D0 F0
90C0- 4C C9 DE 68 85 B8 68 85
90C8- B9 A2 01 B5 6D 95 3E 06
90D0- 1E 26 1F B5 50 48 CA 10
90D8- F2 BA 8A 18 E9 F4 49 FF
90E0- 48 20 87 92 18 65 3E 85
90E8- FE A5 3F 65 1F 85 FF A5
90F0- FE 69 02 85 FA A5 FF 69
90F8- 00 85 FB A8 A5 FA 20 E3
9100- D3 A0 00 A5 3E 91 FE C8
9108- A5 3F 91 FE A2 01 A0 0A
9110- 2C A0 09 B5 B8 91 3E 88
9118- 88 B5 75 91 3E 88 88 B5
9120- FA 95 FE B5 73 91 3E B5
9128- 6F 95 73 CA 10 E3 88 B9
9130- 68 00 91 3E 88 D0 F8 18
9138- 68 A0 0B 91 3E 69 0C 65
9140- 3E 85 1E A5 3F 69 00 85
9148- 1F 20 03 92 90 21 20 14
9150- 93 A5 08 B0 02 F0 0D 08
9158- 20 91 92 28 F0 EB 20 85
9160- 92 4C 49 91 A2 2A 4C 12
9168- D4 4C 30 E4 4C D5 E8 A0
9170- 00 B1 17 F0 04 C9 3A D0
9178- F3 A9 02 20 87 92 A2 01
9180- B5 1E D5 FA D0 E3 CA 10
9188- F7 A5 09 A0 00 91 3E A5
9190- FD D0 08 A5 FE 85 FC A5
9198- FF 85 FD 20 D3 91 A2 05
91A0- B5 FA 95 69 CA 10 F9 68
91A8- 85 75 68 85 76 68 68 A0
91B0- 0B B1 3E F0 08 AA C8 68
91B8- 91 3E CA D0 F9 4C D2 D7
91C0- AF CF 4C 49 4E 4B A5 6D
91C8- C5 3E D0 98 A5 6E C5 3F
91D0- D0 92 60 A2 01 B5 B8 B4
91D8- 17 95 17 94 B8 CA 10 F5
91E0- 60 A2 05 BD EC 91 95 00
91E8- CA 10 F8 60 4C 3C D4 4C
91F0- 3A DB A5 00 F0 04 C9 4C
91F8- D0 04 45 03 F0 ED A2 25
9200- 4C 12 D4 20 B7 00 F0 7B
9208- C9 2C F0 0B C9 CB D0 04
9210- A5 07 F0 45 4C C9 DE A5
9218- 07 D0 48 20 B1 00 49 28
9220- 85 08 D0 16 20 B1 00 A5
9228- FF 48 20 7B DD 68 85 FF
9230- 20 B8 DE A9 40 85 12 4C
9238- C6 91 E6 09 20 E3 DF 20
9240- C6 91 A0 00 A5 9B 91 1E
9248- C8 A5 9C 91 1E C8 A5 81
9250- 91 1E C8 A5 82 91 1E 38
9258- 60 C6 07 A5 FE 85 FC A5
9260- FF 85 FD A9 40 85 14 20
9268- B1 00 49 28 85 08 D0 03
9270- 20 B1 00 20 E3 DF A9 00
9278- 85 14 A5 08 D0 C4 20 B8
9280- DE 38 60 18 60 A9 04 18
9288- 65 1E 85 1E 90 02 E6 1F
9290- 60 A5 07 D0 51 18 A9 07
9298- 65 FE AA 48 A9 00 65 FF
92A0- 48 A8 8A 20 E3 D3 A5 08
92A8- D0 1F 18 A5 FE 69 02 85
92B0- 85 A5 FF 69 00 85 86 A5
92B8- 11 D0 08 A5 12 20 63 DA
92C0- 4C D4 92 20 7B DA 4C D4
92C8- 92 A0 06 B1 9B 91 FE 88
92D0- C0 01 D0 F7 A0 01 A5 82
92D8- 91 FE 88 A5 81 91 FE 68
92E0- 85 FF 68 85 FE 60 A0 02
92E8- 18 B1 9B 65 FE 48 85 94
92F0- C8 B1 9B 65 FF 48 85 95
92F8- A8 A5 94 20 E3 D3 A0 02
9300- 18 B1 9B 65 9B 85 96 C8
9308- B1 9B 65 9C 85 97 20 9A
9310- D3 4C D4 92 20 D3 91 20
9318- B7 00 18 F0 36 A6 07 10
9320- 0B C9 CB D0 3F 20 B1 00
9328- 46 07 10 03 20 BE DE A5
9330- 11 48 A5 12 48 20 56 93
9338- 68 C9 40 F0 07 C5 12 F0
9340- 03 4C 76 DD 68 C5 11 D0
9348- F8 A5 08 F0 05 A0 02 20
9350- 4E 92 38 4C D3 91 A2 00
9358- 20 B7 00 86 10 85 81 20
9360- 7D E0 B0 03 4C C9 DE 86
9368- 11 86 12 20 B1 00 90 05
9370- 20 7D E0 90 0B AA 20 B1
9378- 00 90 FB 20 7D E0 B0 F6
9380- C9 24 D0 06 A9 FF 85 11
9388- D0 0C C9 25 D0 0F A9 80
9390- 85 12 05 81 85 81 8A 09
9398- 80 AA 20 B1 00 86 82 60
93A0- 86 06 A5 02 C9 D4 08 20
93A8- B1 00 F0 1D C9 30 D0 B4
93B0- 28 F0 10 A2 01 B5 04 95
93B8- 73 B5 AF 95 69 CA 10 F5
93C0- 20 E1 91 20 B1 00 4C 6C
93C8- D6 28 F0 26 20 F2 91 A9
93D0- FF 85 86 20 65 D3 C9 B0
93D8- F0 18 38 A5 69 E9 02 AA
93E0- 85 17 A5 6A E9 00 85 18
93E8- C5 B0 90 06 D0 07 E4 AF
93F0- B0 03 4C 79 D9 A0 00 84
93F8- 10 18 B1 17 85 FE 69 0C
9400- 85 1E C8 B1 17 85 FF 69
9408- 00 85 1F A0 0B B1 FE 20
9410- 87 92 A0 00 B1 FE 85 09
9418- A5 6D 85 3E A5 6E 85 3F
9420- A5 1E C5 17 D0 06 A5 1F
9428- C5 18 F0 7B A0 03 B1 1E
9430- 85 82 88 B1 1E 85 81 88
9438- B1 1E 99 CE 00 88 10 F8
9440- A5 09 D0 4E A9 40 85 14
9448- 20 69 E1 A9 00 85 14 A0
9450- 02 B1 9B D1 CE D0 12 18
9458- 65 9B 85 96 C8 B1 9B 65
9460- 9C 85 97 B1 9B D1 CE F0
9468- 05 A2 78 4C 12 D4 A0 02
9470- 18 B1 9B 65 CE 85 94 C8
9478- B1 9B 65 CF 85 95 18 A5
9480- 9B 69 02 85 9B 90 02 E6
9488- 9C 20 9A D3 20 85 92 4C
9490- 20 94 20 4F E0 C6 09 20
9498- C6 91 A0 06 B1 9B 91 CE
94A0- 88 C0 01 D0 F7 F0 E5 A2
94A8- 01 66 06 A0 0A 2C A0 09
94B0- B0 04 B1 FE 95 B8 88 88
94B8- B0 04 B1 FE 95 75 88 88
94C0- B1 FE 95 73 B5 FE 95 6D
94C8- CA 10 E3 88 B1 FE 99 68
94D0- 00 88 D0 F8 A2 F8 9A A0
94D8- 0B B1 FE F0 0C 18 69 0B
94E0- A8 B1 FE 48 88 C0 0B D0
94E8- F8 A5 69 C5 01 D0 09 A5
94F0- 6A C5 02 D0 03 20 E1 91
94F8- 20 95 D9 4C D2 D7 03
```

Varspace Source Code

For those who wish the *Earl of Varspace* source code, it is available in three forms: If you belong to *The Source*, you can download it from the exclusive *Call –A.P.P.L.E.* section, you can subscribe (see *Bytes from A.P.P.L.E.)* to our new disk of the month, or it will be on *Anthology* diskette 11A, available about 3/1/84.

Computer Software Associates, an established software house, supplies software for APPLE, COMMODORE, and IBM home computers. CSA is currently looking to expand its product base of serious programs in the areas of home, education, and office.

Computer Software Associates offers over 30 years of market experience, worldwide distribution, innovative packaging, generous royalties, and programmer support.

Interested programmers should contact Mike Mahan, Product Development Department, Computer Software Associates, 50 Teed Drive, Randolph, Massachusetts 02368 Tel. (617) 961-5700.

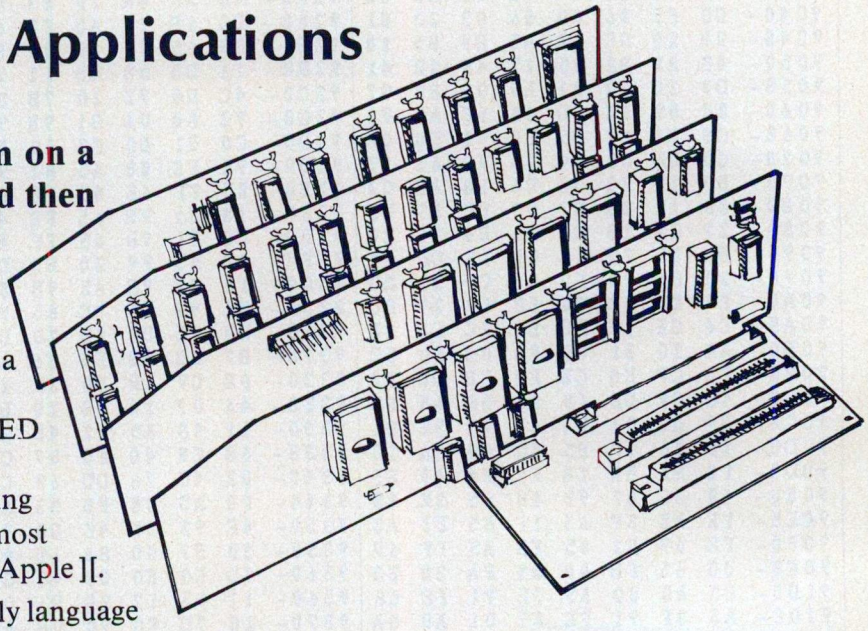
COMPUTER SOFTWARE ASSOCIATES

Circle 12 on Reader Service Card

APPLESEED T.M.

Apple Compatible Hardware for Dedicated Applications

Design and program a system on a standard Apple computer and then market it with the low-cost APPLESEED system . . .



Designed for use in systems where a full Apple computer may not be necessary or desired, the APPLESEED Bus system retains all the essential features of an Apple system, including mother board compatibility with almost every plug-in board available for the Apple][.

Programs are developed in assembly language on a standard Apple][. Upon completion, a set of EPROM's are blown and inserted into the APPLESEED Bus system, which then takes control of program execution.

You buy only what you need for your application; gone are the built-in keyboard, the consumer-oriented packaging and graphics subsystems. Cost, size and power consumption are substantially reduced while flexibility is greatly increased. Compatibility with a multitude of sophisticated programming and hardware add-ons, widely available for the Apple][, has been fully retained, providing a wide range of options for designing customized systems. (Applesoft PROM's are not included.)

A sampling of APPLESEED Bus components:

CPU BOARD The 6502 has been removed from the mother board and included in this card, which also contains the clock circuitry for the system.

64K RAM BOARD 64K of dynamic RAM with provision for control to be used in the normal Apple][conventions; deleting COOO-CFFF and software control over the DOOO-FFFF area.

EPROM BOARD Jumper selectable for 2716 (EPROM) or 34199 (PROM); blow 2716 assembly language programs and then run them in this EPROM board. Jumpers allow selection of software on/off, initialize on/off and other controls.

UART BOARD Software/hardware select all standard EIA baud rates. The board supports four 6850 UART's which allow four independent input and output EIA lines. Each UART is fully programmable for parity, bit length, baud rate, etc.

UNIVERSAL BOARD The APPLESEED Universal board contains the following: One 2716 socket for an F8 PROM, one 2K × 8 bit RAM one UART channel for serial communications, 8 parallel INPUT lines (TTL), 8 parallel OUTPUT lines (TTL), device decode for one additional standard slot-dependent board, support for standard Apple][style keyboard, and a power supply connection. Each feature is independent and any combination may be used simultaneously. The Universal board is designed to be used with the APPLESEED CPU board.

TEXT BOARD The TEXT board is designed to add the conventional TEXT (page 1 and page 2) to an APPLESEED system. The board generates composite video to plug into a standard monitor or T.V. modulator. The board provides 40 columns × 24 rows in each page and is not slot dependent.

MOTHER BOARD 10-slot motherboard is fully compatible with most Apple add-on cards currently available.

Please call or write for prices and more detailed information on these and other APPLESEED Bus products.

DOUGLAS ELECTRONICS 718 Marina Blvd. • San Leandro, CA 94577 • (415) 483-8770

SPECIAL APPLICATIONS:

The Festive Apple

Burt Webb

THERE are two major events held each year at the Seattle Center. The oldest is the Folk Life Festival which has been held annually on Memorial Day Weekend since 1970. The second is Bumbershoot which has been held annually on the Labor Day weekend since 1972. The Center comes alive with artistic exhibits, crafts, food vendors, concerts, films, dancers, jugglers and many other activities. The Northwest Folklife Festival is sponsored by the Folklore Society. The Festival is the largest free regional folk festival in the country. This year, there were more than 700 performances on 15 stages spread over a four day period.

In 1982, my friend, Ed Goehring, became the production manager for the Festival. He held a paid position and was part of the core staff that begins planning the Festival each year in January and works throughout the winter and spring on preparations. In 1982, he asked me to come on as the technical manager. I began working with the Festival several weeks before the Memorial Day weekend. The event was a great success and the technical end ran very smoothly. This year, I decided to use my Apple][computer to help plan the event. In mid-April, I arrived at the Seattle Center to fulfill my role as technical manager and computer consultant for the 1983 Northwest Folklife Festival.

I have owned an Apple computer since 1977. My Apple has serial number 2330 and was one of the first Apples sold in Seattle. I first met Val Golding in the fall of 1978 at an early A.P.P.L.E. meeting. My membership number is #18. I have watched the fantastic growth of the industry in general and our users group in particular and I know I was right to buy an Apple][. I have expanded the memory and installed a new power supply. The addition of Applesoft in ROM, extra colors, and an autostart ROM turned my system into an Apple][Plus. In 1980, I started my own business in microcomputer consulting.

One of my early uses for the Apple was word processing. I used the old Applewriter program which is easy to learn and use. The major problem with using the Apple][for writing is the lack of lower case. I checked out options and got a Lazer lower case chip for $40. I installed it myself by pulling a chip or two in the keyboard driver section and got lower case. The lower case chip worked fine, but I got tired of using [ESC] to capitalize letters. I removed the lower case chip and replaced it with a Videx Keyboard Enhancer for $125. I had to install it in place of the keyboard encoder card. This gave me upper case with the standard shift keys and made the keyboard operation closer to a regular typewriter. In order to get Applewriter to work with the Keyboard Enhancer, it was necessary to graft a short utility program onto the Editor.

I then decided I needed 80 columns. After checking out the various 80 column boards, I bought a used Videx board from a friend for $240, since I already had the Videx Keyboard Enhancer. I now had my upgraded word processing hardware but needed a program which could make use of the 80 columns. I had worked with Applewriter by Paul Lutus and decided to try his new package, Applewriter II. You can easily transfer old Applewriter files

highlight

to Applewriter II. Applewriter II works immediately if you have the Smarterm 80 column board which is supported by Apple. If you have a Videx board, you have to get a pre-boot disk from Videx for $20 to use Applewriter II. There was no problem with compatibility, but it does take a moment to boot the first disk and then put the Applewriter II disk in the drive.

A word processor is worthless without a printer to create the actual paper document. I have used an Epson MX-80 dot matrix printer for years. I purchased the version which has both tractor feed and friction feed for single sheets. I had an early model which did not have graphics capabilities, so I bought a set of Graphtrax ROMs and installed them. One of the MX-80 features I like is its ability to define different font options. You can do enhanced type which fills in the spaces between the dots on the letters and makes them look more like typewriter fonts. Then there are the specialty fonts, italics and double striking for boldface. You can have double width or compressed mode which allows 132 columns per page. In order to use all these fancy options with the old Applewriter, I had to get an E-Driver utility program from a Eugene, Oregon firm and graft it onto the Print program. Applewriter II has the ability to embed printer control codes in the text without using this utility.

The type from one of these full feature dot matrix printers is quite readable and almost indistinguishable from some typewriters, but it is not "letter quality." We could use the Epson for most purposes at the festival, but there were a few documents such as the Event Service Order for the Seattle Center Staff which had to be done on a letter quality printer. For the needs of the festival, I rented a Transtar 130 from Anchor Computers in Pioneer Square in return for an ad in the Festival program.

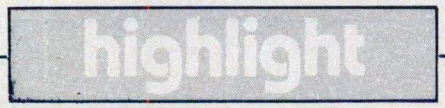
During the planning for the festival, many different letters had to be sent to performers, volunteers, exhibitors, food vendors, etc. Often there were paragraphs which were common to different groups. The computer allowed a "cut-and-paste" operation where paragraphs could be assembled and special text added to create a particular letter. With minor revisions, these letters can be used each year. Each different group required detailed procedures for the festival. We developed the standard procedures for each group this year and created packets which were mailed or handed out before the festival. The computer was invaluable in the creation and revision of all the different forms, procedures, letters and contracts that were necessary to set up the framework for the festival.

When a major event is held at the Seattle Center, many different departments cooperate to supply the necessary services. The creation of the stages is handled by the stage crew. All sound equipment is taken care of by the sound department. The custodians do all the cleaning and placement of chairs. The engineers deal with the physical buildings and plumbing. All the wiring is done by the electricians. Security has responsibility for all the locks on the doors and gates, as well as parking on the grounds. All of these departments must know when they are supposed to do something and where. This is accomplished by a document called the Event Service Order.

Each department has a section which spells out in detail the actions to be taken in each of the locations around the Center and exactly when they take place. The number of stages must be decided along with the maximum size of the audience. The maximum number of microphones must be determined for each stage. The layout of the crafts area and food vendors area must be decided upon and mapped. The opening and closing times of all the stages must be specified. Many, many details have to be to be determined and recorded. The ESO is in a process of continual revision up to the last moment. Before this year, this complex and time consuming process had to be done manually on typewriters.

With the Apple][, the process was simplified and speeded up. Revisions could be added easily. Whole sections could be moved around at will. Applewriter II can search for all occurrences of a specific word and, if desired, replace them with another word automatically. With the text-embedded print commands in Applewriter II, we could shift the left margin in and out simply by typing ".lm20" and ".lm5" at appropriate places in the text. This was necessary to place abbreviations for the different facilities on the left of the text so they would be easy to see at a glance. The document was so long that we had to break it up into four sections to accommodate the 30,000 character working space of the program. Fortunately, Applewriter II is equipped with a feature that permits the starting page number to be set for each file printout so we were able to have sequential page numbers for the whole document. Next year the process of creating the ESO will be even simpler because we can start with this year's ESO and just make the changes on the wordprocessor.

We used the Visifile program to create useful data bases for the Festival. Visifile is a good program which has small menus for each function. These are displayed in the top two lines of the screen and you step through them with the left and right cursor arrows. Using Visifile, you define your own forms by using a 15 character name for each field. You choose whether you want an alphanumeric field (sort from left to right) or a numerical field (does calculations and sorts from right to left). You can have 28 fields containing a total of 255 characters per record. The total number of records you can put on a disk depends on the size of the individual record. Each disk can hold around 125,000 characters so you can put about 500 records on a disk if they use the whole 255 characters allowed. You are able to do multiple sorts up to 10 fields deep and each file can have up to 24 sort indexes. Reformatting is allowed in case the file structure needs to be modified or you want to merge two files. You



create a new file format, match the fields in the new format to the fields in the old format, and shift the data over. You can chose to shift only the data that meets your criterion.

Visifile reports can be formatted and generated in a list structure. If a field has five letters, you only get five letters at the top of the column for the name of that field. We found that the easiest way to handle this was to be sure that we didn't use a name that was longer than the maximum length of the field. We had to do a lot of creative abbreviating. Any or all of the fields can be printed out in any order desired.

The volunteer files posed some real challenges. We had about 300 volunteers during the festivals. The shifts were each three hours long and there were 12 different shifts. There were about 12 areas of responsibility including production crews, traffic, security, souvenir sales, information booths, etc. Each area might require up to 10 people per shift. People usually signed up for three shifts at most and stated their three preferences for activities. Key sort index cards have been used in the past, but they are time consuming and you have to type out all the information for the report.

This year, the data was entered into the Apple][as the forms came in. We created a system for sorting on the preferences against the time slots to create a list of available people for each of the shifts and tasks. What we attempted was to sort first by each shift then by task preference and finally alphabetically by name. People kept volunteering up to the last day before the festival so there was never a time when we had all the volunteers in the data base in time to do a full series of sorts for every shift and task before the festival. The volunteer coordinator had to deal with people changing their minds on shifts and preferences up to the last moment. She had to create and update a huge wall chart because she had limited access to the computer.

We had over 700 performances scheduled at 15 staging areas over a four day period. While the volunteers could be coordinated on a huge wall chart and assigned during the festival, the performers had to be sorted by time and place in time to be entered into the program so it could be printed a week before the festival. A form was created and mailed out that recorded the vital information about the performers such as name, address, brief description called a tag line, special requirements, etc. A huge grid was created for the whole festival and the performances were filled in as they were scheduled. This process went on for weeks until just before the festival.

We created the performers file structure and entered records for each performance. This form was much shorter than the mail out and contained the name, tag line, time of performance, place of performance, etc. Because the performers were being scheduled as a continual process, we set up the data base with slots for all the scheduled performances and left empty fields for the names of the performers and the tag lines. As performances were scheduled, they could be entered in the data base. Just before the festival when all the performances had been scheduled and the data base was complete, it was time for the sorts.

One interesting problem that appeared involved day and time. We had performances scheduled for 11 AM and 1 PM, for example. In a numerical sort, 1 would appear before 11. In alphanumeric sorts, the numbers come after the letters, so 1 PM would still be sorted out before 11 AM. The same sort of problem arose in the day sorts. We were operating on Friday, Saturday, Sunday and Monday. Monday would be sorted out ahead of Saturday and Sunday. We solved this by creating a special field for sorting that never appeared on the printout. We called this military time because we used the 24 hour clock that they use in the military. In this system, 1 PM became 1300 hours and sorted after 11 AM which is 1100 hours. We also included the date to take care of the problem with the days. Friday was the 27th, Saturday was the 28th, and so on. By combining the date and military time into one number, we had a system which would correctly place all of our times in sequence. For example, Saturday at 1 PM would become 281300 and Monday at 11 AM would become 301100.

After all the information was entered, we did one sort that started with the time of the performance, then alphabetized the stages and finally gave the name of the performer(s) and their tag line. This was distributed to all the information booths so they could tell people what was happening all over the grounds at a specific time. Then we did a sort that started with alphabetical sort of the stages, then the performance times and ended with the name of the performer(s) and the tag line. This sort was used by the typesetter to lay out the program. It was also used by the stage managers and by the sign makers to create the daily performance schedule signs for each stage. Another sort started with an alphabetical listing of the performers, then sorted on time of performance and finished with stage. This was useful to people that wanted to find the performance time and place of a particular favorite group. These different types of listings had been done by hand during past festivals if the organizers had the time.

Last year, I took the ESO and all of our production tasks and spent three days creating a time script for the festival production. It was very time consuming, difficult to integrate last minute changes and almost indecipherable to anyone else. This year, I started by creating a file format for most of the tasks in the ESO. The Center staff had to carry out these tasks but we were responsible to check that they had been done properly and on time.

highlight

THE FESTIVE APPLE

The form I created had to be very open because the format of the information changed from task to task. About all I could do was create several long fields of 65 characters each, that were called Task/Action. For example, "Check 6 microphones and monitor speakers." We included the standard time and date and also included the date/military time trick for sorting. We needed the standard abbreviation for all the facilities at the Center as well as the full name of each. There was a reference to the page number in the ESO and also any diagram that might be needed to detail the location of something. We also needed to specify which department carried out the task, such as custodians or stage. I spent several afternoons entering about 250 separate items from the ESO.

At the same time, the production manager was creating a form for our own tasks. This differed slightly from the ESO form. We had to have fields to identify who had requested the task and the urgency of the task. These tasks consisted of things that our own production crew had to do for the festival. We kept entering tasks up to the day the festival started. The total number of our own work orders was about 150.

One of the nice features of Visifile is the ability to reformat data files. You create a new file definition and shift records from the old format to the new format. We created a combined definition that satisfied both the ESO and the Work Orders format and shifted both sets of data to the new format. Then we used the merge function to combine both files into one file. The Visifile program can store about five hundred 255 character records on one Apple 3.3 disk. Our combined file held a little over 400 entries so we still had plenty of room.

The most important sort was a time sequence. We sorted by date first, then by hour and finally by the location. This was the combined record of everything the Center staff had to do and everything the festival production staff had to do on a minute by minute basis for the four days of the festival. The next sort was for the individual spaces. This was very useful to keep track of all the activities in a particular space. Some of these depended on other actions so we had to check to make certain that they were carried out in proper sequence. This would have been difficult from the time sort alone. This sort started with the location, then went to the date and finally to the time.

We spent hours trying to figure out the best format for the printout. We wanted to have the abbreviation for the facility, the facility name, the date and the time by themselves on the left side of the page. Then we had to include space for the other bits of data such as the page number of the ESO, the requesting party, who had to carry out the task and so on. Finally, on the right side of the page we wanted the two 65 character lines of Action/Task. The fact that each of our lines of text was considerably longer than 80 characters posed a problem. Since we were working with a dot matrix printer, we could use the condensed print and get 130 characters to the line. We had to use every available space to crowd everything in.

Visifile has a special configuration section that includes space for special printer control codes. We had already tried to use the space for a control code that called the enhanced print on other printouts. The command was an ESC-E, but the Visifile would not recognize an escape character or any of the alternative representations such as CHR$(27). We had to use a small utility program to set the enhanced print before we booted Visifile. When it came time to call for the condensed print, the command was an ESC-P. We could not use this, but there was an alternative. SI (Shift In) could also call the condensed print. Visifile would not recognize this either so we used [Ctrl-O] in place of SI (See page 138 in the Applesoft Manual). Visifile required the use of # to stand for Control so we placed # O in the printer format line and got condensed print. We created a dozen copies of each of the printouts and distributed them to the production teams.

We now had our operations script and it was our bible for the five days of the Festival. We had three production teams which kept in touch on FM walkie-talkies. These were invaluable during the festival. The center grounds cover many square blocks and we had 30 different areas of activity to worry about. Things started slowly on Thursday when we laid out the exhibition areas for crafts and the food vendors area. The tempo increased on Friday as the crafts and food vendors moved in and the stages were set up. We had to deal with over 400 exhibitors and food vendors and their vehicles.

The festival officially began on Friday evening. By noon on Saturday, the festival was in full swing. Each day, there were certain changes of the space layout for different activities. Some facilities were used for concerts part of the time and then were rearranged for dancing. We had to double check to make sure these spaces were changed back and forth at the right times. Each day we had to check the set up of the outdoor stages and the opening of the facilities that housed activities. Each evening we had to clear out spaces and make certain they were locked up.

One by one, the items on the operations script were checked off. I put in 15 hour days. I saw perhaps one or two hours out of the more than 300 hours of performances. I did get a chance to spend an hour looking over the crafts. On Monday, things got really crazy after the festival officially closed and we had to move all the crafts and food vendors out. There was a lot of confusion with traffic keeping me there 'til midnight. The festival was finished for another year and I was totally exhausted.

It took me a week to recover from the festival. I left my computer down at the office to assist in the paperwork that had to be finished up. I picked it up and used it to draft my 12 page evaluation report. There were a lot of disks generated in the course of the festival. I am glad that there are two big events each year at the Center since I didn't get a chance to see

(continued on page 31)

highlight

SPECIAL APPLICATIONS:

COMPUTERIZING SCIENCE EXHIBITS

Dave Taylor

Introduction

FIVE years ago when *Call -A.P.P.L.E.* was still a small users group newsletter being written in Val's basement and copies made on a photocopy machine, the Pacific Science Center started its involvement with the Apple Computer. Those of you not from the Pacific Northwest may need some introduction to what the Science Center is and what we do. The Pacific Science Center is a non-profit science and technology museum located in Seattle next door to the Space Needle in the six building complex that served as the US Science Pavilion in the 1962 World's Fair. We currently have over 650,000 visitors each year coming to see our 200 plus exhibits and displays and to participate in our specialized education programs. It is in some of these exhibits and programs that we have grown to depend on the Apple computer as a valuable tool.

Very Early Days

In 1978 the Science Center purchased its first Apple with serial number 3148, one of the early models that didn't have Blue or Orange High-Res colors, and Applesoft wasn't even available on Cassette tape, much less in ROM. I soon learned that there were other people in town with Apples that had formed a users group and some of these people even knew what was going on inside the Apple. I knew that I needed to join and thus became member number 221 of what has become the 10's of thousands of members of A.P.P.L.E.

Using this Apple and a cassette recorder (until the then still to be released disk drive arrived some months later), we began exploring the possibilities for creating an exhibit that would allow the public to get their hands on the still mysterious and scary computers. I don't know if many of you remember what a "glorious" experience it was to use cassette storage and lose program after program or wait five minutes for a program to load only to get the reward of "ERR" on the screen.

Computerworks

With the help of two grants from local foundations, 10 Apple][systems with disk drives and color monitors were purchased and a computer exhibit/lab known as Computerworks was established. We decided that we wanted to allow visitors to interact unattended with science and technology related programs during the normal exhibit hours. We also wanted to be able to use the computers to teach programing in BASIC during the evenings. We converted one of our classrooms for this dual purpose and built cases to protect the Apples from vandalism.

During the day the computer programs for the visitors use include...

- Space Maze — Teaches the idea of building a program
- Medicine — Includes subprograms: First Aid, High Blood Pressure, Alcohol, Facts About You
- Science Games — Contains subprograms: Know Your Sciences, Towers of Hanoi, Shooting Stars
- Astronomy — Contains: Ast. Crossword, Solar System Quiz, Planets, Lunar Lander
- Human Cannonball — Explores basic projectile ballistics
- Metric Mixup — Contains: Metric Quiz and Science Mixup (Anagrams)
- Trap Apples — Contains: Trap, Sell apples, Guess the State; all dealing with limits
- Perception — Contains: Short Term Memory, and What's It?

Video system

An ever present problem at the Science Center is letting the visitor know what events are scheduled in our six buildings. It seems that no matter how many signs are put up there are not enough or they need to be changed. In an attempt to help this problem, we borrowed an idea from airline terminals and developed a computerized signage system. An Apple with disk and clock card, located in our central offices runs ten black and white TV's located throughout the buildings which display times of films, demonstrations and other special events.

—→

highlight

COMPUTERIZING SCIENCE EXHIBITS

If a time must be changed or an event canceled, a few keystrokes keep the TV displays current. The system cycles through a listing of the events of the day listed with times and locations, and graphic screens advertising upcoming events. If a child is lost, a message can be displayed instantaneously letting the parents know where to find their child. This system has not eliminated our need for temporary and permanent signs, but we have found it helps.

Other Computerized Exhibits

Computers have been used in a variety of other exhibits in the Science Center including programs on weather, land use, air quality, water utilization, energy and nutrition. A program located in a new exhibit allows the visitor to enter information on their home and lifestyle and receive an analysis of the present energy utilization, along with ways to make their home more energy efficient. A number of programs have been developed on nutrition under grants from the Washington State Dairy Council.

These programs allow the visitor to explore their eating habits and nutritional knowledge, one allows the visitor to select a sample meal and receive an analysis of the vitamin, calorie and mineral content. A set of curriculum materials have been developed and extensive testing and revision of both the program and the materials have gone on during the past year prior to our making them available for purchase through the National Dairy Council.

The Users

The user of computers in exhibits at the Pacific Science Center demands special attention in the design of software. Most computer software is made with assumptions that we can't make with the average visitor using our exhibits. First, that the user has had some previous experience with the computer being used; second, that the user has read at least a minimum amount of instructions before using the software or has been shown by someone else what to do; and third, that the user is not trying to break the machine and/or the program.

The visitor has only a few hours to visit the entire Science Center and in general can't be expected to spend more than 5–10 minutes at any one exhibit station. Our programs must allow the visitor to learn what to do and have a successful interaction very quickly. Instructions are limited to one or two screens and the object of the program must be simple, clear and direct. In general we plan our programs to take 3 to 5 minutes to complete at which time the visitor can try again or move to another program. This also serves to prevent one person from hogging the computer for too long while others wait.

Our programs must also be able to recycle to the beginning if someone abandons them part way through the program, so we build timers into our input routines and if a key has not been pushed in 30–45 seconds the program starts over and waits for the next visitor.

We knew when we first planned to have computers running unattended on the exhibit floor that we would have to make the programs as "bulletproof" as possible. Simple things like an alpha answer to a numeric input will cause a program to bomb, so will "[Ctrl-C]," RESET, and other simple visitor responses. With proper care in developing input routines some of these can be avoided, the use of the Applesoft ONERR routine catches others and more recently the "Autostart" monitor ROM allows revectoring of the RESET key. Early on, the "Old" monitor ROM and Integer Basic didn't have these tricks so we had to physically cut the connections on the reset and control keys.

Ron and Darryl Aldrich, the authors of Higher Text, were local high school students at that time and found great joy in finding ways to break into our programs and then teaching us how to prevent others from doing the same. Our computer exhibits have the Apples in some form of protective case with the keyboard exposed, but even so we have had incidents of vandalism. I have found on a number of occasions, kids (some over 18) pounding their fists on the keyboards, we have had keycaps stolen, and on two occasions have had individuals break into the case and remove chips. The Apples have fared pretty well with this type of use.

hardware

Problems with Pets

The Science Center through other grants was able to purchase 24 Commodore Pet computers for use in another computer lab and in exhibits. At the time the Pet had the advantage of being considerably cheaper than the Apple and was being used widely in educational circles. Unlike the Apple this has proved to be a short term solution as we are now looking for funding to replace many of these computers as they have worn out through use and are not as easy to repair as the Apple.

The lack of color graphics and the fact that they are no longer a system supported by the manufacturer means that the public isn't as anxious to take classes on these as with the Apples. Commodore released at least three versions of BASIC during the last five years, none of which is compatible with the preceding version (different POKEs, PEEKs and CALLs to the system). Thank goodness Applesoft and DOS have not gone through these changes, although I understand that the new version of DOS called ProDOS to be released early next year will be substantially different.

Easily Fixable Apple

Unlike our experience with the Pets, the Apples have proven to be much more reliable. Ninety percent of our problems have resulted from corrosion of contacts on peripheral cards and on chip leads. This is usually solved by reseating the card or the chips. Individual keys failing have been our second most prevalent problem, understandable since they get hundreds of thousands of keypresses each year and an occasional pounding by some of our not so mature visitors. The early switching power supplies were not as well built as the later ones and we have had to replace a half dozen of these. All in all, once you learn the few quirks of the Apple it is fairly easy to keep our more than 40 machines going.

Computer Van & Camp

The summer of 1982 was exciting for us as donations from a number of the hi-tech companies in the Seattle area allowed us to purchase 15 more Apples for use in a Computer Day Camp. Groups of 28 students met daily for six hours in one week sessions to learn introductory programming in BASIC. Six of these one week sessions were held in 1982 and 10 sessions were held this past summer.

These same 15 Apples are now used during the school year for a travelling Computer Van program. During its first year of operation, more than 9000 students were contacted in these sessions where a team of two Science Center teachers set up the computers in a school classroom and gave up to eight 45 minute introductory computer literacy lessons during a one day visit. The demand for this program has been so great that we will be adding a second computer van during this coming school year. The only negative aspect is that the teachers have to haul 15 Apples and 15 monitors back and forth between the van and the classroom each day.

Computer Fair & Apple Festival

The Pacific Science Center has been the site of an annual personal computer fair co-sponsored by the Northwest Computer Society since 1979, and this year's took place on the weekend of November 11th - 13th. For the past two years we have also worked with A.P.P.L.E. to sponsor an Apple-Festival weekend here, introducing the public to the capabilities of the Apple Computer.

Program Development
The goals

In evaluating any exhibit technique we have to take care to match the goal of educating and entertaining with the best media to get the message across to the visitor, this is also true with computers. We use the following criteria to evaluate if it is appropriate to use a computer.

- Does the use of a computer help to personalize the visitor's experience or relate the subject matter to the visitor's life?
- Will the use of the computer allow the visitor to manipulate variables, make decisions, and/or see information in a new way?
- Will the use of a computer program allow the visitor to gain access to more information on the exhibit subject matter in areas of special interest?
- Can the computer program be used to interpret or augment a real object or experience elsewhere in the exhibit?
- Will using a computer motivate the visitor or a different portion of our visitor audience, to spend time in the exhibit that they wouldn't otherwise?

Types of programs

The programs we have developed fall into a number of basic categories...

- Quizzes — "Test Your Knowledge" & vocabulary building programs.
- Simulations — Manipulate variables and see the results.
- Games — More mind puzzles than "shoot-em-up," hand-eye coordination games in our case.
- Data Bases — Let the visitor review information on a particular subject or object, such as weather data for any day of the year.
- Decision Making/Problem Solving — Similar to simulations, make a hypotheses and test the results.
- Personalized Information — including the Nutrition and Energy Programs I mentioned above.
- Interpreters of Real Objects or Experiences — Display temperature or pressure in an exhibit, or graph over time, etc.

Programming Style

When we first started developing programs for the Apple, we followed the classic method of starting at the beginning and continuing to write code until we had reached the end, letting the program develop as we came to coding it. After the first few instances where we had to revise programs after being away from them for a few months and when someone else had to try to modify programs, we decided it was time to find better methods. Over time, we have developed a structured programming style common to most of our programs with a mainline routine from which all other routines are merely subroutines to be filled in. The mainline routine is essentially an outline we can write for the whole program. For example...

(continued on page 31)

highlight

SPECIAL APPLICATIONS:

HIGHER TEXT IN ACTION

Steve Brugger

THE Pacific Science Center is a technology and science museum in Seattle, Washington. In our exhibits program, we use over 30 microcomputers, most of them Apples. Most visitors to the Center stay over two hours and try out a large percentage of over 200 exhibit devices. This translates into only a few minutes that a visitor spends at each device.

In the five years that we have been using microcomputers on the exhibit floor, we have learned by trial and error what makes an effective learning experience for our "typical" visitor. Effective use of the Apple's graphic capabilities is one of the sure-fire ways to catch and retain the visitor's interest. At the Science Center, we've gathered a library of useful programs and utilities that are used in nearly every program we write. One of the best is Ron and Darrell Aldrich's Higher Text, one of the earliest and finest programs from A.P.P.L.E.

Because of the number of programs and computers we have running and our small staff, we've learned (often painfully) that new programs being developed must be standardized to a high degree, to make maintenance and revisions simpler and less time consuming. Higher Text lends itself to this standardization, but because of its length and location, it has imposed some memory limitations on the programs we write.

All of our Apples used as exhibit devices are 48K][+'s. Since our programs depend heavily on High-Res graphics, we often use both High-Res pages. Specialized input routines (The routine figures out when someone has walked away and starts the program over automatically and one input routine we have used is called a "smut filter." You really don't need an explanation do you?), sound effects, light pen routines and fast disk read routines all make demands on the available memory. Since the programs are used by several hundred thousand people a year, we refrain from heavy disk access. Spending Monday mornings recopying crashed disks is a lousy way to start the week!

To make the most of 48K memory, we have ferreted out many of the available nooks and crannies in Higher Text and put many of our machine language routines in there. Now if we could only keep track of all our versions of Higher Text!

One of the first available spots is the space taken up by the Integer Basic routines at locations $800 through $C00. Since we're running][+'s, these aren't needed. Into this spot has gone a routine to load High-Res pictures directly through the DOS read-write-track-sector. With this program in place, High-Res pictures can be loaded in about 2.5 seconds. To use this program, however, the pictures must be copied onto the disk before any other programs as they must occupy two contiguous tracks.

Another large area of memory available is that space taken up by the large font, at locations $1400 to $1C00. Higher Text stores a complete small font character set and a large font, upper case only, in its standard implementation. This gives you the option of using more than one font style in a program, but since we rarely use the large font, this space is often used to store other machine language routines. Don't be too quick to dump the large character font, however, because these characters can be redefined to allow surprisingly fast and intricate block animation. Space Maze, a program written by one of our volunteers, uses the large fonts of Higher Text quite effectively to provide fast, flicker-free animation from within a BASIC program.

In one program that is on the exhibit floor, we use light pens for some of the inputs. This is a simple photo-transistor circuit that uses one of the switches on the game paddle port. The original light pen and routine came from a commercial source, but we have modified both the hardware and software for our purposes. The light pen can scan up to 32 squares on the screen in about three seconds. The modifications we made to the hardware were mostly to beef up the casing to protect it from heavy use. (What do you think your ballpoint would look like if 50 different people chewed on it every day? Yechhh!)

The software was more extensively modified to allow the light pen to be used with Higher Text. With these modifications, the squares that are scanned can be normal sized or tall, wide or expanded fonts. They can also be different colors, though we've found that white usually produces the best results. After an initialization call, the routine only needs the location of each of the squares on the screen to be scanned and then a call to begin the scan. We were able to store this routine where the Integer High-Res routines were.

Yet another area of memory available for routines occurs between the end of the Higher Text routines and the beginning of the large font table. This is located from $1272 to $1380. Higher Text does not use this area for any purpose that I have found.

We've standardized the prompts and error messages that a visitor sees on the video screen to allow them to concentrate more on the subject. These prompts will always be in a particular font, with a particular color. To allow these to be used from within a single subroutine, the current color and font information is first PEEKed, then stored as temporary variables. The values are then POKEd back into those locations when the subroutine is complete. The location of the color value is $302. The value there corresponds to the Higher Text color. The font information is stored at location $30C. A zero represents regular font, $C0 represents the expanded font and so on. If you plan to use boldface in your programs, this flag is stored at $311.

—

highlight

HIGHER

(Continued from page 30)

Along with Higher Text, comes the &LOMEM: utility, which is very handy in its own right. The directions for its use are very clear in the Higher Text manual if you want to move your program upwards in memory. For some of our menu driven programs however, we have had to move programs downward in memory. To do this, you must first change the values at locations 103 and 104. These two locations are the low and high bytes for the beginning of BASIC pointers. This pointer tells the BASIC interpreter where the first line of BASIC code is. You must also POKE zeros into the first two locations where the new beginning of the BASIC programs will be. If you wanted to move from 24586 to 16384, you would first have to put zeros in locations 16384 and 16385. Then POKE the value 64 (16384/256) into location 104. The low byte pointer at 103 does not need to be changed if you remain on a page boundary.

```
1000  REM MOVE NEXT
      PROGRAM DOWN IN
      MEMORY
1010  POKE 16384,0: POKE
      16385,0
1020  POKE 103,1: POKE 104,64:
      REM BEGINNING OF
      BASIC PROG.
1030  PRINT CHR$(4)"RUN NEW
      PROGRAM"
```

This will mess up your variables, however, so don't use it in the middle of a program, only when moving from one program to another.

If &LOMEM: is used only at the beginning of a program, then other routines can be put in this space once the BASIC program has been moved. &LOMEM: resides at $330 to $3CF.

Writing this article has had me thinking about some of our first attempts to put characters on the High-Res screen. We used shape tables, constructed by hand. That was back in the dark ages, four or five years ago. I wonder where we'll be five years from now?

FESTIVE

(Continued from page 26)

any of this festival while I worked on it. I really enjoy doing big events because of the excitement, the challenge, and the people I work with. I always look forward to Bumbershoot when I can attend and be one of the crowd.

I received many compliments for the technical operations of this year's festival. Of course, the success of the festival depends on good team work, with some of the credit for technical operations belonging to my Apple][. I spent much of my time dealing with the few problems that arose. As far as the performers, exhibitors, and audience was concerned, this year's festival ran very smoothly. The staff appreciated the contributions of the Apple system, and I was happy to be able to bring together two of my main interests; event production and microcomputers.

Planning for next year's festival is already starting. The Festival has purchased a microcomputer with a hard disk. With the files we can use from this year, there will be much less data entry required. Because we have the computer all year, we will be able to prepare the programs, formats and sorts far in advance. The whole process should be smoother, faster, and more comprehensive.

I look forward to a time when the evolution of microcomputers will put the memory and abilities of the Apple in the palm of my hand. Then I will be able to carry around the operations script in electronic form. It could be updated each morning from a master file in the office computer. There could even be a radio link to the office computer so I could call up lists of participants, parking permits, background information, etc. If there was a phone patch, I could call out to other places and access data bases. With the evolution of graphic display technology, I could call up displays of floor plans. No matter how sophisticated these event production technologies become, I will always remember that it all started with my good old Apple][.

SCIENCE

(Continued from page 29)

```
100  REM Program Name
110  REM Author, Date, etc
120  REM Short Description
...
200  REM MAINLINE ROUTINE
210  GOSUB 1000: REM Setup
     Routines
220  GOSUB 2000: REM Graphic
     to Attract Visitor
230  GOSUB 3000: REM Has key
     been pressed
240  IF no key pressed GOTO 230
250  GOSUB 4000: REM Instruc-
     tions
260  GOSUB 5000: REM Next task
....
Other task routines...
....
290  GOSUB 9000: REM Play
     Again?
300  IF yes then GOTO 260
310  GOSUB 9500: REM Thank's
     For Playing
320  GOTO 220: REM Await next
     user
```

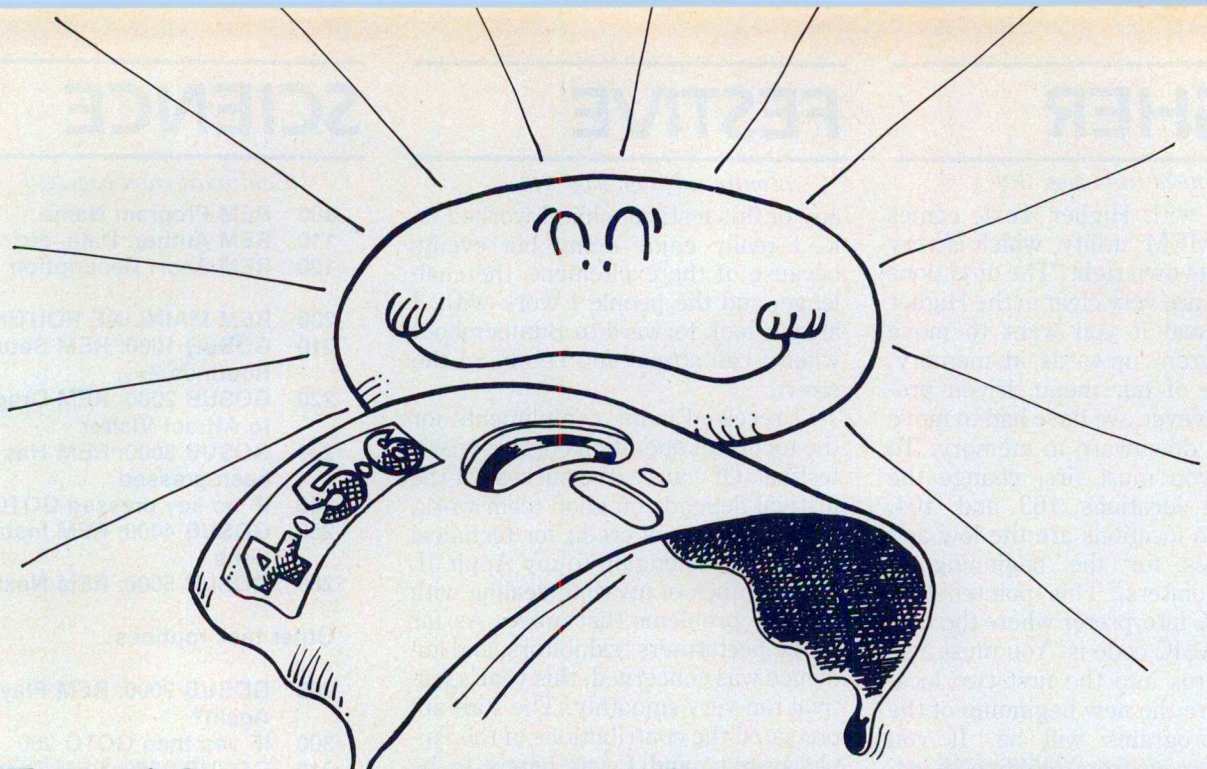
The use of REM statements and clear variable names is important in our style. In general our programs are not pushing for great speed as they are not shoot-em-up arcade style games, so the use of REM's isn't a problem; where speed is necessary we usually will write a machine language routine that is called from the BASIC program.

Summary

Although only a few copies of most of our programs exist, I would venture to guess that they have been used by more individuals than most commercially available software due to our large number of visitors each year. The public have become more sophisticated in their knowledge of computers since 1978. Very few people are seeking to learn the BASIC language, but instead are looking for information on how to use computers as a useful tool for work, home and education.

The computer has proven to be a valuable tool in our work in exhibits and our education programs. We feel it will continue to increase in importance in passing along knowledge and experience on science and technology.

highlight



What if . . . your software gets too soft?

You need software insurance.

Diskettes are fragile, and when a protected program is damaged, the results are expensive and inconvenient. If you have a backup diskette, though, you can have your Apple, IBM or compatible computer back on line within seconds . . . affordably. That's software insurance.

Copy II Plus

This is the most widely used backup program for the Apple. Rated as "one of the best software buys of the year" by **InCider** magazine, its simple menu puts nearly every disk command at your fingertips. The manual, with more than 70 pages, describes protection schemes, and our **Backup Book**™ lists simple instructions for backing up over 300 popular programs. The Backup Book is expanded bimonthly, and is always available to Copy II Plus owners for only $1.00 (and a self-addressed, stamped envelope). Best of all, Copy II Plus is still only **$39.95.**

WildCard 2 (formerly The Alaska Card)

Designed by us and produced by Eastside Software, WildCard 2 is the easiest-to-use, most reliable card available. Making backups of your total load software can be as easy as pressing the button, inserting a blank disk and hitting the return key twice. WildCard 2 copies 48K, 64K and 128K software, and, unlike other cards, is always ready to go. No preloading software into the card or special, preformatted diskettes are required. Your backups can be run with or without the card in place and can be transferred to hard disks. **$139.95** complete.

Important Notice: These products are provided for the purpose of enabling you to make archival copies only. Under the Copyright Law, you, as the owner of a computer program, are entitled to make a new copy for archival purposes only, and these products will enable you to do so.

These products are supplied for no other purpose and you are not permitted to utilize them for any other use, other than that specified.

Copy II PC

This is **THE** disk backup program for the IBM PC, PC/XT and PCjr that copies almost anything. Others may make similar claims, but in reality, nothing out performs Copy II PC . . . at any price. Copy II PC even includes a disk speed check and is another "best buy" at only **$39.95.**

We are the backup professionals. Instead of diluting our efforts in creating a wide variety of programs, we specialize in offering the very best in backup products. So, protect your software investment, **before** things get too hot.

CENTRAL POINT Software, Inc.

The Backup Professionals

To order, call 503/244-5782, 8:00-5:30 Mon.-Sat., or send your order to: Central Point Software, 9700 SW Capitol Hwy, Suite 100, Portland, OR 97219. Please include $2.00 for shipping and handling.

A Call -A.P.P.L.E. LANGUAGE REVIEW:

"C" What I Mean? (Part Two)

Jim Hockenhull/reviewer

Aztec C Compiler $199
Manx Software Systems
Box 55
Shrewsbury, NJ 07701
Not copy protected

THE first installment of this article began by making a distinction between a programming language as an abstract, formal entity and its down-to-earth implementation on a real computer. Later, we saw that the distinction may not be as clear-cut as theoreticians would have us believe. I'd like to continue in that vein a bit before getting to the point of this article: a review of the Aztec C compiler for the Apple][.

What is C? (III)

C grew up with UNIX; UNIX grew up with C. Both grew up at Bell Laboratories with the PDP-11 series of minicomputers by Digital Equipment Corporation (DEC). While UNIX, C, and the PDP-11 are not inextricably welded together, they do relate closely to one another. Transporting C to another environment seems to require an implementation which emulates the original situation in which the language was created.

Back in Part 1, we said that an implementation was actually a group of programs performing certain functions. These programs tend to have distinct identities in a compiling system. They are: text editor, preprocessor (in C's case), compiler, assembler, linker, and command interpreter. It's likely that these terms are unfamiliar to the BASIC programmer, so let's examine them a little more closely as they apply to C.

The *TEXT EDITOR* accepts input from the terminal and outputs a file of C source code interspersed, perhaps, with commands to the preprocessor. The UNIX system requires that the name of this file ends in ".c," as in "prog.c".

The *C PREPROCESSOR* inputs the source file and obeys whatever applicable commands it finds there, expanding macros, including other source files, and substituting strings. Its output is the expanded C source code.

The *C COMPILER* converts this source into a file of assembly language code, specific to a particular machine. The *ASSEMBLER* then translates this into relocatable object code which can be installed at any memory location.

The *LINKER* joins such relocatable modules into a file of executable machine code. The modules may have been written by the user or may exist in libraries. In UNIX, the file created by the linker is automatically named "a.out." The programmer then renames it to suit. The executable file is run by invoking its name from the keyboard.

Presiding over all of this is the *COMMAND INTERPRETER,* a program allowing communication with the functions above and with the computer itself. The UNIX command interpreter is called the "shell".

If we have written a C program via the text editor and saved it as a file named "prog1.c," then the command:

```
cc prog1.c
```

entered from the keyboard, will invoke the entire series of programs necessary to compilation, ending with the creation of the executable file "a.out". Options are provided to permit the suppressing of parts of the compilation procedure, the joining of several source or object modules into a single program, and the linking of libraries.

All of the above is specific to the UNIX operating system where C is usually found. I have dwelled on it at some length so we can see later how C's normal environment has been modified for the Apple][.

A Sample Program

Let's look at an actual C program (Listing 1). The simple but important task of this program is to print a string to the screen. Though trivial, Listing 1 shows us several things about C. It consists of a definition of a function called **main**, which requires no arguments as indicated by its empty argument list, ( ). The body of **main**( ) is enclosed by curly braces; it consists of only one statement, a call to another function, **printf**, followed by a semicolon. **Printf** is a formatted output function found in the standard C library. It does take arguments, in this case the literal string "Hello, world!" We can safely guess that **printf** will print its argument to the screen or some other output device. But what's that stuff at the end of the string? "Backslash-n" is C's representation of the non-printing "newline" character. BASIC will automatically put out a newline at the end of a print statement unless it is suppressed by a semicolon. The reverse is true in C: no newline will be printed unless it is explicitly part of the output. C provides shorthand for all of the non-printing ASCII characters.

```
/*  LISTING #1                          */
/* a simple C program.                  */
/*                                      */

main()
{
   printf("Hello, world!\n");
}
```

"C" WHAT I MEAN (Part Two)

On a UNIX system we would save the program, probably as "sample2.c," compile it by the command:

```
cc sample1.c
```

which would also link the **printf** function from the standard library, and finally run it by typing:

```
a.out
```

the name of the executable file.

But we're not using UNIX, we're using Apples. At long last let's see how Manx Software Systems has made the necessary adaptations.

The Aztec C Compiler

Package Description:

The Aztec C Compiler System exists entirely as software. Six 5¼" floppy disks contain: a command processor, two different compilers, a 6502 assembler, a pseudo-code assembler, a linker, several versions of several libraries, and a number of utility and device-support functions. The disks are not copy-protected. A loose-leaf manual provides some 140 pages of documentation.

The system requires an Apple][/][+ with 48K of main memory, a 16K expansion card, DOS 3.3, and one or two disk drives. No additional hardware is needed, though support is provided for several brands of 80-column cards.

System Description:

All of the requisite components of a compiler system, as described earlier, are present, some in several versions. let's look at them individually.

• The *COMMAND INTERPRETER,* reflecting its UNIX heritage, is called the "shell." It is designed not to replace DOS but to superimpose itself over the DOS image, which must be present in memory. This means you must boot one of your own DOS disks before BRUNning SHELL. The Aztec C disks will *not* boot by themselves.

The shell provides the control and escape sequences necessary to enter lower case and other characters not available on a standard Apple. It renames most DOS commands with shorter UNIX-like names: **ls** ("list") for CATALOG, **rm** ("remove") for DELETE, and so forth. Unlike DOS, shell commands can take multiple file names as arguments. For example:

```
rm file1 file2 file3
```

will delete all three named files from the disk. A very handy command, **cp**, will copy a file from one disk to another.

Of course, the Aztec shell is *not* UNIX. Its efforts to look like UNIX may occasionally confuse or irritate those familiar with the larger operating system. The command **cd**, for instance, means "change directory" in UNIX but stands for "change disk drives" in the Aztec shell (which has no "directories").

When a string of characters is entered from the keyboard, the shell will attempt to interpret them as one or more shell commands. Failing to do so, it will assume that the string is a file name and will search the disk(s) for such a file. If it finds the file *and* the file is a text file, it will open that file and try to read it as one or more shell commands. (This allows you to save often-used sequences of shell commands as disk files, rather like DOS's EXEC files.) If the shell finds the file *and* the file is an executable binary file, such as a fully-compiled C program, the file is — well — executed. *If* the file is of neither type *or* no file of that name is found, appropriate error messages are emitted.

```
program1 > pr:
```

will load and run the file "program1" and send its output to the file "pr:", which just happens to be the printer driver.

• The *COMPILERS*: The world of computers is a world of seemingly unavoidable compromises or "trade-offs." One of the most common is that between the amount of space a program takes up in memory and its running speed. Stated simply, the faster a program is, the more space it takes up; the more compact a program, the slower it will run. Aztec C offers the programmer several degrees of compromise by providing two different compilers, two assemblers, and two sets of libraries.

Now let me make one thing perfectly clear (to coin a phrase): a language translator, such as a compiler, takes a *source* language as input and outputs an *object* language. Though "object code" is usually taken to mean "machine code," it ain't necessarily so: object code can be any language. The Aztec C compiler named "cci" translates C source into assembly code for a machine that doesn't exist — a pseudo-machine. The assembler called "asi" then converts this code into instructions for the non-existent machine — pseudo-code or "p-code" which, in turn, must be *interpreted* at run time into executable 6502 code. Interpretation, of course, takes time. The cci compiler was the only one supplied with early versions of Aztec C.

Now a second compiler, called "c65," is part of the system. It converts C source directly into 6502 code ("native code") which does *not* need time-consuming interpretation at run time. Native code runs from 5 to 20 times faster than p-code, so obviously the c65 compiler is the way to go, right?

Well, not quite. You see, the faster native code also takes up about twice as much memory space as the slower p-code. The old trade-off.

Segments of programs can be translated with either compiler and the resultant modules can be linked into a complete program, so that time-critical parts can be compiled into native code with c65, others into space-saving p-code with cci. When running speed is absolutely important, modules can be written directly in 6502 assembler. Such modules are easily interfaced with higher-level C routines. The programmer is free to decide when and where to make the time vs. space compromises.

• The *LINKER*: Output from the compilers or assemblers takes the form of *relocatable object code* which can operate from any location in memory. Relocatable modules are linked together with specified libraries into a coherent whole by the linker, named "ln." Its output is a file of executable machine code which can be run by entering its name from the keyboard.

As a very simple example of how this all works, let's assume we've created a file of C source code named "prog1.c," and that we wish to compile it into native (6502) code. We enter:

```
c65 prog1.c
```

causing the shell to locate and execute the c65 compiler with the argument "prog1.c." Compilation produces a disk file of relocatable object code named "prog1.rel." It's a safe bet that our program will make use of some standard library functions, so we must link it to the library named "sh65.lib" by typing:

```
ln prog1.rel sh65.lib
```

If all is well, the result will be an executable file named "prog1" (no suffix). Entering this name from the keyboard will cause the file to be loaded and run.

This is about the simplest use of the system. The p-code compiler/assembler is slightly more cumbersome. In all cases options are available to suppress stages of the compilation process or to create different names for output files.

• The *LIBRARIES*: We saw earlier that C is heavily dependent on libraries of pre-defined functions. Aztec C includes the standard C library, a library of floating point routines, and a "stand-alone" library which allows the creation of programs which will run on any Apple, without the presence of the Aztec C system.

Each of the three libraries comes in two versions, one compiled into native code, the other into p-code. Again, choices can be made about the space/time trade-off by selecting which library to link into the program.

• *ARCHIVES* are files of the C source code for the libraries and utilities. They may be examined by the user and are a rich trove of programming information.

• The *TEXT EDITOR*: Source files are written using the text editor "ved." (I assume this stands for "visual editor"; UNIX editors have names like "ed" and "fred.") Ved is a screen-oriented editor: text is accessed by moving the cursor around the screen. Line numbers exist for reference only.

BASIC programmers may have trouble remembering to enter the "insert" mode to add text and then to exit to the "command" mode in order to move on, but that's the way most text editors (including Applewriter) work.

Like the shell, ved provides ways to enter lower case and other non-Apple characters. The control sequences used for this take some heavy memorization. What gets printed on the screen may not resemble the character it represents. But C programs can be written on a regular old Apple without any additional expense.

By Manx Software's own admission, ved is "not a particularly fast or smart editor but it gets the job done."

Users Comments

First, as to the language C itself: it is not a universally admired language. It has been called a computer expert's language, written, developed, and used by experts. Indeed, the very features that make it so powerful — system-level access, pointers, preprocessor commands, and so on — may serve simply to make it confusing to the non-expert. Still, I feel that those who are serious about writing programs would benefit from some sustained contact with C. At the very least, it would make them better BASIC programmers.

Aztec C offers a reasonable means for Apple users to have that contact. Manx Software claims that their product is a complete implementation of UNIX Version 7 C, the current version, with the exception of an obscure feature called "bit-fields." So far I've run into nothing to dispute that claim. This *is* C, folks, or something awfully close to it.

The complexity of the task of transposing C down to the 8-bit environment is attested to by the six floppies that make up the system. At least four of those disks are going to be swapped in and out of the drives almost constantly. (Here let me dispute Manx's claim that you can operate Aztec C with a single disk drive. In my opinion, two drives are essential and, were the system set up for it, three or four would be better. This is a *big* software package.)

When C is moved to an Apple is it good for anything? To get some idea of performance, I wrote a version of the "Sieve of Eratothsenes," an algorithm often used to as a "benchmark" for checking running speeds. (See Jim Gilbreath, "A High-Level Language Benchmark," *Byte,* September, 1981.) Source code for the program is given in Listing 2. Translated into p-code, the Sieve took about 45 seconds to run; compiled into native 6502 code it took just over 9 seconds. By comparison, an equivalent Applesoft program required 4 minutes and 55 seconds. These figures show quite clearly the price paid for BASIC's interactive friendliness.

When compiled into a "stand-alone" module, a binary file which can be BRUN on any Apple without the Aztec system present, the C Sieve took up about 15K of memory; the Applesoft version needed a little over 300 bytes. A hands-down victory for BASIC! — until you remember that Applesoft lugs around something like 10K of interpreter, without which you can't run the 300-byte program. I suspect, however, that Applesoft's space advantage would increase as programs get larger.

It is only through writing large programs that I would feel qualified to pass a definitive judgment on the usefulness of Aztec C. But the system throws enough petty frustrations in the way that I have not wanted to tackle lengthy projects, nor do I want to suggest that anyone run out and buy the package for the sheer fun of it.

"C" WHAT I MEAN? (Part Two)

It's not fun. C is a complex language and Aztec C is a complex implementation. There's a lot that has to be explained and understood. The documentation rises to its formidable task but, unfortunately, comes out the loser. Much does not get explained — or understood.

For one thing (and Manx Software makes no bones about it), the manual is strictly about the Aztec C system; it is *not* about C itself. Aside from a single tutorial chapter, the best part of the document, it assumes detailed knowledge of the language. Any general C questions are referred to Kernighan and Ritchie's *The C Programming Language,* the standard work on the subject. But this little volume is itself deceptive. Though written in a light, friendly style, it is definitely by experts, for experts. Armed with only the Aztec manual and K & R, the non-expert may soon wish he'd spent his 200 bucks on adventure games.

With the exception of the very good tutorial chapter, the manual spends most of its time bogged down in treacherous tarns of hacker-ese such as:

> "If the '-s' is specified as a compile time option and a pointer is to a member name that is not defined in the structure associated with the pointer then all previously defined structures will be searched until the specified member is found."

That sentence actually says something; the manual actually says something. You must simply be willing to scrabble the meaning out by brute force and persistence.

The manual says a lot of things, but does not say everything, especially about the little bugs and quirks that still plague Aztec C. At present, for example, using Version #1.05B of the system, I am unable to compile C source with the cci p-code compiler. Compilation always aborts with an "OUT OF MEMORY!" error. If the cause and remedies for this error are in the manual, I haven't as yet been able to locate them.

Generally speaking, if everything goes well, Aztec C is fine to use. But when the unexpected happens, as it does all too often, the documentation offers little or no help.

Conclusion

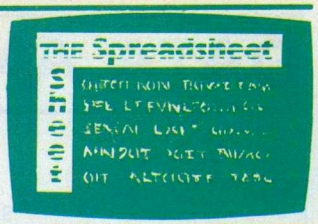
C is powerful enough and in wide enough use to make it worth experiencing. Aztec C is an ambitious attempt to render C workable on an Apple. Its designers have taken ingenious measures to overcome the inherent differences between the needs of C and the facilities of the Apple][. As yet, the system contains enough flaws to be frustrating, although Manx Software Systems seems genuinely concerned about improving their product. It *is* C; as such it offers a reasonably-priced opportunity for the Apple user. The nature of the Aztec C system and its documentation, coupled with the nature of C itself, will make the going, if not rough, at least highly challenging.

Definitely *not* for beginners.

```
/*  LISTING #2                          */
/* Sieve of Eratosthenes                */

#define TRUE 1
#define FALSE 0
#define SIZE 8190
#define SIZEPL 8191

char flags[SIZEPL];

main()
{
  int i, k, prime, count, iter;

  printf("1 iteration\n");
  for (iter = 1
;iter <= 1;iter ++)
  {
    count=0;
    for(i = 0;i <= SIZE;i ++)
      flags[i] = TRUE;
    for(i = 0;i <= SIZE;i ++)
    {
      if(flags[i] == TRUE)
      {
        prime = i + i + 3;
        k = i + prime;
        while(k <= SIZE)
        {

  flags[k] = FALSE;
            k += prime;
        }
        count ++;
      }
    }
  }
  printf("%d primes\n", count);
}
```

THE SPREADSHEET SHEET:

DATA ANALYSIS (Part Two)



Jeffrey Jacques, Ph.D.

Introduction

IN the last article we began our exploration into the world of data analysis. We examined an efficient way to summarize a variety of variables by using some of the available Spreadsheet functions. We noted how to invoke @MIN, @MAX, @AVE functions and how to construct a worksheet to compute the additional summary statistics of variance, standard deviation, skewness, and kurtosis.

These summary statistics may be sufficient for some of your needs, but you will likely find yourself with the additional need to summarize, display and report data in tabular form. This may be accomplished by using simple and grouped frequency distributions, and bar graphs.

Contrary to what some peopl may have told you, you can creat these types of tables with electronic spreadsheets. All that is required is some forethought, the construction of an efficient worksheet, and your A.P.P.L.E. *Spreadsheet* software. Once the basic worksheet is created, we'll be able to simply insert the necessary rows and columns, substitute current data, modify a few cells, and easily recompute/perform tabular analyses with other data sets.

The functions we will use in the construction of this tabular worksheet are:

1. @IF
2. @AND
3. @SUM
4. @COUNT
5. @INT
6. @ROUND

Some General Thoughts on Building Tabular Displays

Any good tabular display must include: (1) a title, (2) appropriate labels for columns and labels for values — where appropriate, (3) a count of the number of occurrences of each value or contiguous range of values, (4) a running cumulative percent distribution, and (5) appropriate summary totals.

Since there are no conditional @COUNTing routines, the process of creating a simple or grouped frequency distribution must be done in several steps. First, identify the number of different values. Second, for each subject, determine if the subject had that value. Third, @SUM the number of occurrences of that value. Once summed, we may take the last major step and enter the number of counted occurrences in the appropriate table.

Before we examine the faculty worksheet and begin the construction of our first tabular display, it is useful to review the codebook and note the summary statistics computed last time. Go ahead and get these tables. I'll wait . . .

Modifying the Worksheet: A Simple Frequency Distribution

Although some people may like to create a separate template for working with frequency distribution tables, I find it easiest to create such tables in an unobtrusive area of the worksheet. We shall need, at least for the variable of rank, ten rows (i.e., one for the title, two for column labels, one for each unique value, one for the total line, and three for drawing lines to distinguish among different parts of the table.) Using the worksheet from last time, we will use rows 28 through 38.

In columns M through Q we shall formally display the results. For example, once we count the number of occurrences of each value, we will re-enter this value next to its appropriate label and immediately under the "Cases" column. The PCT. value, at least for the first value, may be computed by the formula:

```
@ROUND(O33/O38*100,1)
```

If you are using VisiCalc™, then try:

```
@INT(((O33/O38)*1000 + .5))/100
```

That is column O, rows 33 and 38. These little routines are helpful for rounding off values when the number of cases divided by the total number is not an integer. The "CUM PCT" or cumulative percent column was computed by @SUMming in each higher step the total number of cases to that point, and using the same rounding off routine as listed above. Therefore, for the first cumulative percent value — instructors, we can use the same equation as that listed above. Beginning with the value of 2 — assistant professor, the equation is:

```
@ROUND(@SUM(O33 . . . O34)
/038*100,1)
```

For the third ordered value, the appropriate formula is:

```
@ROUND(@SUM(O33 . . . O35)
/038*100,1)
```

and so on until all discrete higher order values are handled.

In the lower half of Table 2.1 you will find all this information and more. Take a few moments, load your worksheet from last time, and create this simple frequency distribution tabular display. Bye the way, judicious use of the /Reproduce command will shorten your work considerably. Just be careful which cells are Relative and which should be No Change.

Okay, you've probably noticed a substantial number of @IF statements in the top half of Table 2.1. We will use them to count the number of occurrences of each value. It would have been nice (Are you listening Bill Graves?), to be able to use a conditional @COUNT function, such as:

```
@COUNT(@IF(column = value),
B4 . . . B24)
```

or, even better, a conditional sum function, such as:

```
@SUM(@IF(column = value,1,0),
B4 . . . B24)
```

—●→

business

DATA ANALYSIS (Part Two)

TABLE 2.1

SPREADSHEET TEMPLATE FOR

SIMPLE FREQUENCY DISTRIBUTION OF FACULTY BY RANK

	M	N	O	P	Q
1			Scratch Space		
2					
3	Rank	1	2	3	4
4					
5	3	@IF(M5=1,1,0)	@IF(M5=2,1,0)	@IF(M5=3,1,0)	@IF(M5=4,1,0)
6	1	@IF(M6=1,1,0)	@IF(M6=2,1,0)	@IF(M6=3,1,0)	@IF(M6=4,1,0)
7	3	@IF(M7=1,1,0)	@IF(M7=2,1,0)	@IF(M7=3,1,0)	@IF(M7=4,1,0)
8	3	@IF(M8=1,1,0)	@IF(M8=2,1,0)	@IF(M8=3,1,0)	@IF(M8=4,1,0)
9	2	@IF(M9=1,1,0)	@IF(M9=2,1,0)	@IF(M9=3,1,0)	@IF(M9=4,1,0)
10	4	@IF(M10=1,1,0)	@IF(M10=2,1,0)	@IF(M10=3,1,0)	@IF(M10=4,1,0)
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
24	1	@IF(M24=1,1,0)	@IF(M24=2,1,0)	@IF(M24=3,1,0)	@IF(M24=4,1,0)
25					
26		@SUM(M5...N24)	@SUM(O5...O24)	@SUM(P5...P24)	@SUM(Q5...Q24)
27					
28		Summary of the Variable : Rank			
29					
30					Cum.
31	Label	Value	Cases	PCT.	PCT.
32					
33	Instruc	1			@ROUND(O33/O38*100,1)
34	Assist.	2			@ROUND(@SUM(O33.O34)/O38*100,1)
35	Assoc.	3			@ROUND(@SUM(O33.O35)/O38*100,1)
36	Full	4			@ROUND(@SUM(O33.O36)/O38*100,1)
37					
38		Total	@COUNT(M5...M24)		

Note: In the PCT. column, place the following formulae:

@ROUND(O33/O38*100,1)
@ROUND(O34/O38*100,1)
@ROUND(O35/O38*100,1)
@ROUND(O36/O38*100,1)
@ROUND(O37/O38*100,1)

Column widths are exaggerated so that you may more easily follow the formulae.

Alas, these are not available currently. We will, therefore, have to use separate columns to accomplish our conditional counts. Here's the process. Since the variable of rank had four different values, we will need four columns. We'll add a fifth column and /Reproduce the original values so that we may easily determine if the logic and procedures are correct. In column M, therefore, /Replicate the values of rank, cells B5 through B24 to M5...M24.

In column N, we will count the number of occurrences of the value "1" or instructors. In cell N5, therefore, enter the function:

@IF(M5=1,1,0)

If his/her rank is a "1," then THE Spreadsheet will place a value of 1 in cell N5, if not, then the value of 0 will be displayed. Simply /Reproduce this @IF statement down the remaining rows (e.g., row 6 through row 24). Then, in row 26, place the function:

@SUM(N5...N24)

By summing all the zeroes and ones in column N, we have "counted" the number of occurrences of "1"s or instructors.

Similarly, in column O, you can count the number of occurrences of "2"s — Assistant Professors — by slightly altering the @IF statement (i.e., in cell O5 enter @IF(M5=2,1,0)). In column P, you can count the number of occurrences of Associate Professors by checking to see if the value in column M is "3." Finally, in column Q, count the number of occurrences of "4" — Full Professors. Under each column, simply @SUM the values. THE Spreadsheet will "count" the number of occurrences of each value when you just enter the ! key. Now, re-enter these newly summed values, as displayed in row 26, into your simple frequency distribution table. Now, press the ! key, and watch your Apple do the rest. Check your results with those listed in Table 2.2.

Note: a quick way to transfer values from one cell to another is with the use of the pound (#) command. For example, place the cursor in cell O26. then enter +N26#. If you are using version 2.14 or newer, the numeric value of cell N26 will appear automatically. With a simple press of the RETURN key, the value of N26 will be entered into cell O26. Repeating this procedure will reduce your workload considerably.

A nice serendipitous outcome of developing a simple frequency distribution is that you now have access to two additional summary descriptive statistics — i.e., the mode and the median. The mode is that score with the greatest frequency. For the variable of rank, however, we have no single mode since there were two occurrences of the greatest frequency — i.e., six instructors and six assistant professors.

The second additional summary statistic that is now easily determined is the median. By definition, the median is that value which divides the distribution in half. By examining the cumulative percent column, find that value which includes the 50th percentile, and note the median value. The cumulative percent which first includes the 50th percentile of rank values is the 60th percentile. The median value for rank, therefore, is "2" or assistant professor.

business

TABLE 2.2

SIMPLE FREQUENCY DISTRIBUTION OF FACULTY BY RANK

	M	N	O	P	Q
1			Scratch Space		
2	==========				
3	Rank	1	2	3	4
4	----------				
5	3	0	0	1	0
6	1	1	0	0	0
7	3	0	0	1	0
8	3	0	0	1	0
9	2	0	1	0	0
10	4	0	0	0	1
11	1	1	0	0	0
12	2	0	1	0	0
13	2	0	1	0	0
14	2	0	1	0	0
15	1	1	0	0	0
16	4	0	0	0	1
17	1	1	0	0	0
18	2	0	1	0	0
19	1	1	0	0	0
20	2	0	1	0	0
21	3	0	0	1	0
22	4	0	0	0	1
23	3	0	0	1	0
24	1	1	0	0	0
25	----------				
26	20	6	6	5	3
27					

Summary of the Variable : Rank

	Label	Value	Cases	PCT.	Cum. PCT.
28	==========				
29	----------				
30	Instruc	1	6	30	30
31	Assist.	2	6	30	60
32	Assoc.	3	5	25	85
33	Full	4	3	15	100
34	----------				
35	Total		20	100	

You could have just as easily used any of the other discrete variables and created similar simple frequency distributions, by modifying the number of rows in the table and the number of columns in your computational space. In this case, the number of rows and columns must equal the value of the inclusive range (i.e., range + 1) for that variable when all values are present.

I would be remiss if I did not point out a shortcut. When you have to create several simple frequency distributions, start with that variable with the largest inclusive range. After completing this table, you can easily /Delete the last row which includes the largest value in the body of the table, re-enter the labels, and enter the next smaller — in terms of number of different values — variable. This will shorten your work time considerably. Now, save this modified worksheet.

Working with Grouped Frequency Distributions

Variables with many different values require a slightly different approach. You will need to group the data for variables like income. To do this effectively, you must determine: (1) the number of class intervals you wish to use in your table, (2) the size of each class interval, (3) the lowest and (4) highest value of each interval, and then (5) use an @IF statement with the logical operator @AND to count the number of occurrences of each interval range.

The total number of cases and the range of scores are good reference points to determine how many class intervals. With only twenty cases, we will need five or fewer class intervals. Even with 100 or more cases, 10 class intervals is usually sufficient.

Starting with the lowest value, or a conceptual meaningful low value, determine the size of each class interval with the following formula:

$$\text{Class Interval Width} = \frac{\text{@MAX - @MIN + 1}}{\text{\#Of Class Intervals}}$$

or

$$= \frac{28414 - 8250 + 1}{5}$$

$$= \frac{20165}{5} = 4033$$

or approximately $4,000 per class interval. We can now modify our tabular display to include faculty income class interval ranges and appropriate labels. After /Replicating the income values into column M, your @IF statements for cell N5 should read:

```
@IF(M5<12000,1,0)
```

while the @IF statement for cell O5 reflects the $4,000 class interval width:

```
@IF(@AND(M5> =
12000,M5<16000),1,0)
```

and cell P5 should be:

```
@IF(@AND(M5> =
16000,M5<20000),1,0)
```

Similarly, additional @IF statements are required until the @MAXimum value for the variable is included in the appropriate column. Or, if you know that there are relatively few widely dispersed values at the upper limit of the range, you might use:

```
@IF(M5> = 24000,1,0)
```

After /Reproducing these @IF statements throughout the appropriate columns, the remaining procedures are the same as that outlined for a simple frequency distribution. See Table 2.3.

Although the modal value for a grouped frequency distribution may be found in the same manner as that for a simple frequency distribution,

—◉→

business

Inside Apple

Vol. 1, No. 4

A dot matrix printer that will improve your image.

Meet the Apple® Imagewriter, the newest dot matrix printer for your Apple Personal Computer.

And with all that it has going for it, just maybe the best dot matrix printer on the market.

Take legibility, for instance.

The Imagewriter crams 140 x 160 dots into each square inch. So you get text that's highly readable and high resolution graphics, besides.

And is it fast.

The Imagewriter cruises at an unbelievable 120 characters per second. And that's just in the text mode. It's even faster printing graphics. 180 characters per second, to be exact.

What's more, the graphics dump is up to 60% faster than other comparably priced dot matrix printers. And that makes the Imagewriter fast enough to handle the Lisa™.

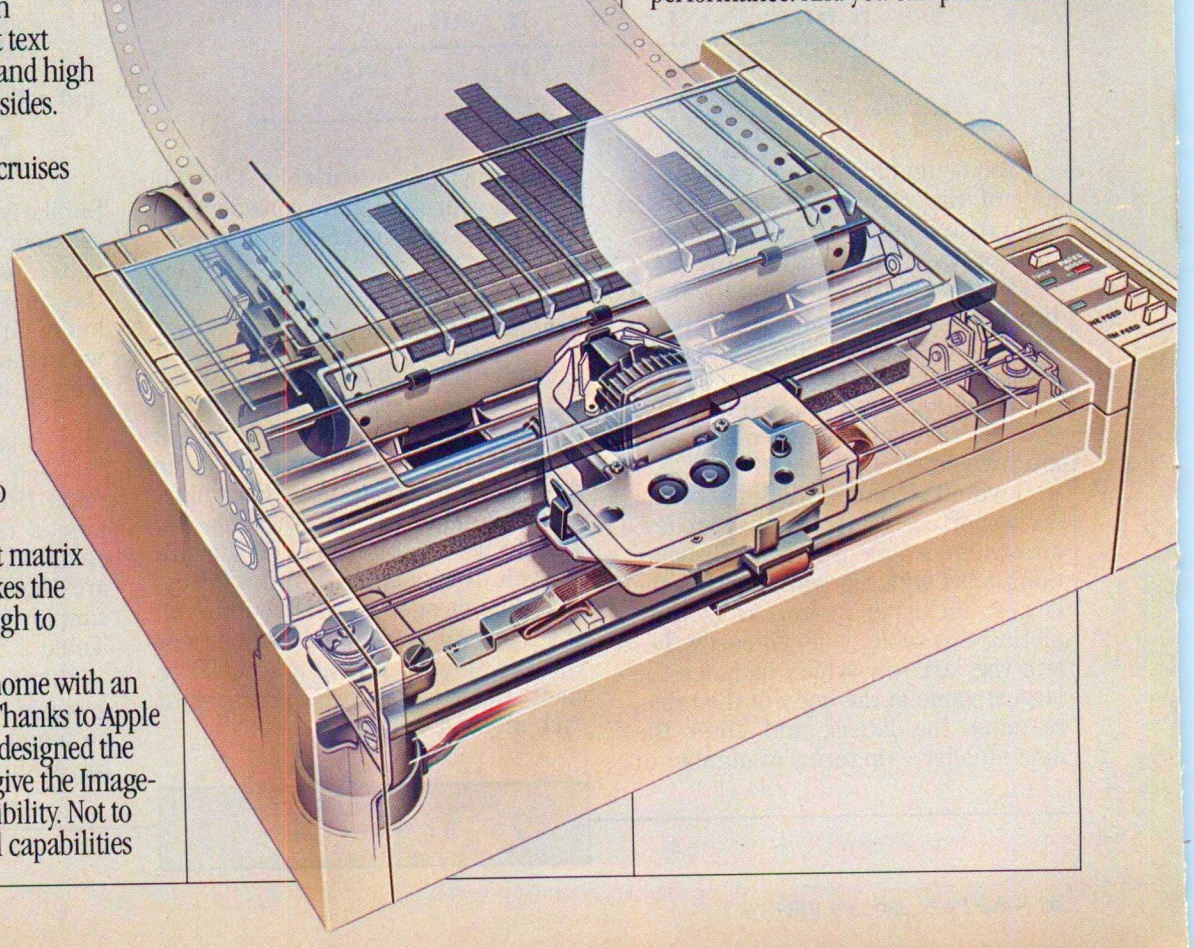
Yet it's just as at home with an Apple III or Apple IIe. Thanks to Apple software experts who designed the control electronics to give the Imagewriter perfect compatibility. Not to mention some special capabilities like superscript and subscript, to name just two.

Now, with all this high-speed performance, you'd expect the Imagewriter to make the Devil's Own Noise. It doesn't. In fact, the Imagewriter is specially constructed — with overlaid seams and special sound-deadening materials — to achieve a remarkable 53 dB. How loud is a remarkable 53 dB? You'd make more noise if you read this aloud.

The Imagewriter even has quiet good looks, since we designed it to look like the rest of the Apple Family.

Yet even with all its improvements, the Imagewriter is a better deal than any other dot matrix printer with comparable performance. And you can print that.

APPLE PRESENTS TH
E IMAGEWRITER APPLE PR
ESENTS THE IMAGEWRITER APPLE PRESENT
S THE IMAGEWRITER APPLE PRESENTS THE IMAGEWRITER APPL



Charge!

Go out there and get the Apple Personal Computer System you really want. Now. Without laying out your extra cash. Without tying up your other lines of credit. With the Apple Card. The only consumer credit card reserved exclusively for the purchase of Apple Computers, peripherals and software.

Like all our products, it works simply:

Fill out an application (short, to the point and annotated in English) at an authorized Apple dealer honoring the Card. Your salesperson will call in the application and in most cases get an approval for you right on the spot.

You can then take your Apple system home. You don't even have to wait for the Card; we'll mail it out to you. And by the time you get it, you'll probably be well into doing whatever you bought your Apple system to do.

There is no annual fee for the Card, although a couple of restrictions do apply. The first purchase must include an Apple Personal Computer and you have to put 10% down. And subsequent purchases need to be at least $100 if made with the Card. Oh, yes — you'll also have a credit limit.

When you use the Apple Card to make additional purchases, all you have to do is show the Card and sign the invoice. As long as it's within your credit limit, of course. Our dealers get a little nervous when someone signs for half their inventory. You understand.

You'll also receive monthly statements that include the latest purchases, credit available, and the minimum payment due. You'll also be happy to know Apple Card credit terms are affordable and the payments can be spread out. It's all spelled out for you at the time your Card is approved.

So stop by a participating authorized Apple dealer and get an Apple Card. Just think of it as credit where credit is due.

Give your floppy disks the boot.

We call it the "floppy disk shuffle." It happens when you have two or more software programs on floppies and you need to work with both. What do you do? You put one disk in, boot it, do your work, take it out, put the other disk in, boot it, do your work — you get the idea.

Well, you can stop shuffling any time now.

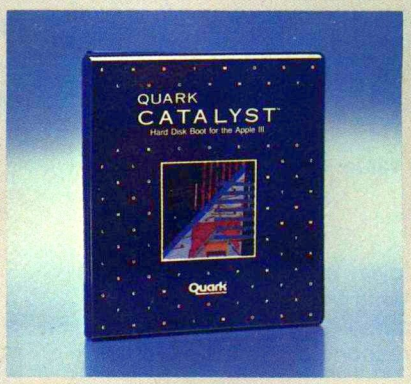
Thanks to a unique new software program called Catalyst™ from Quark, Inc. Specially designed for your Apple III and ProFile™ hard disk.

Catalyst allows you to take a wide variety of software programs and store them on your ProFile. Once they're on your ProFile, you just select the program you want from the Catalyst menu that appears on your monitor — then Catalyst does the rest. You'll never have to boot those programs again.

What kinds of programs will work with ProFile and Catalyst? Almost anything written for the Apple III including copy-protected programs like VisiCalc,® Quick File™ and Apple Writer III. Or languages like Pascal, BASIC, or COBOL.

And once you've loaded these programs into your ProFile, the only diskette you may ever need is the Catalyst.

So if you have an Apple III and a ProFile and more floppies than you care to flip through, get yourself a Catalyst. And boot those disks for good.



Apple Computer Inc., 20525 Mariani Ave., Cupertino, CA 95014. For the authorized Apple dealer nearest you, call **(800) 538-9696.** *In Canada, call* **(800) 268-7796** *or* **(800) 268-7637.** *© 1983 Apple Computer Inc.*

VisiCalc is a trademark of VisiCorp, Inc. Catalyst is a trademark of Quark, Inc. Apple, the Apple logo, Quick File and ProFile are trademarks of Apple Computer Inc.

DATA ANALYSIS (Part Two)

TABLE 2.3

GROUPED FREQUENCY DISTRIBUTION OF FACULTY INCOME

	M	N	O	P	Q	R
1			Scratch Space			
2	==========	==========	==========	==========	==========	==========
3	Income	(12,000	12-16	16-20	20-24	)=24,000
4	----------	----------	----------	----------	----------	----------
5		18442	0	0	1	0	0
6		12167	0	1	0	0	0
7		16858	0	0	1	0	0
8		20641	0	0	0	1	0
9		25000	0	0	0	0	1
10		20970	0	0	0	1	0
11		14775	0	1	0	0	0
12		15441	0	1	0	0	0
13		15595	0	1	0	0	0
14		17940	0	0	1	0	0
15		9525	1	0	0	0	0
16		20690	0	0	0	1	0
17		11865	1	0	0	0	0
18		14381	0	1	0	0	0
19		8250	1	0	0	0	0
20		15000	0	1	0	0	0
21		17117	0	0	1	0	0
22		28414	0	0	0	0	1
23		17424	0	0	1	0	0
24		8494	1	0	0	0	0
25	----------	----------	----------	----------	----------	----------
26		20	4	6	5	3	2
27			Summary of the Variable :		Income	
28	==========	==========	==========	==========	==========	==========

29					Cum.	Bar Graph
30	Label	Value	Cases	PCT.	PCT.	(Cases)
31	----------	----------	----------	----------	----------	----------
32	($12,000	8500	4	20	20	****
33	$12,000-$15,999	12500	6	30	50	******
34	$16,000-$19,999	17500	5	25	75	*****
35	$20,000-$23,999	22500	3	15	90	***
36	=) $24,000	27500	2	10	100	**
37	----------	----------	----------	----------	----------	----------
38		Total	20	100		

(e.g., the modal value in this case is $12,500), the determination of the median value requires slightly more effort. The basic definition for the median is the same as that used with a simple frequency distribution. You must work, however, with class intervals rather than specific values and then estimate the median score from the appropriate class interval. The basic formula is:

$$\text{Median value} = LL + \frac{\frac{\text{Count}}{2} - F}{f} * i$$

where:

LL = lower limit of the interval contains the median value
F = cumulative frequency corresponding to this lower limit
f = the number of cases in the interval containing the median value
i = the class interval width

Substituting the data for the variable of income:

$$\text{Median} = 11{,}999.5 + \frac{\frac{20}{2} - 4}{6} * 4000$$

$$\text{Median} = 11{,}999.5 + 4000 = 15{,}999.5 \sim \$16{,}000$$

Just choose an appropriate cell (e.g., cell Q39) and enter the formula:

```
11999.5 + (((O38/2-O32)/O33)*4000)
```

Note, however, that the formula may change for each variable. In this case it is not surprising that the formula yielded the upper limit of the second class interval since it corresponded to to 50th percentile.

The computations of modal and median values now complete all the measures of central tendency typically reported by any computer statistical package.

Bar Graphs

To create the bar graph, as presented in Table 7, simply set the column width to 10, set the /Format specifications to *, and /Replicate the values in Column O, rows 32 through 36 to column R. This works fine as long as the number of cases is relatively small. When the number of cases is relatively large, you may use the equation:

```
@ROUND(O32/O38)*10,0)
```

This will round off the value to a number between 0 and 10. (Of course, you could have chosen a value other than 10 for this column width. When you do this, simply substitute *10 with the same value as that of the new column width.) By placing this equation in column R next to each appropriate row value, setting the column width to 10, and setting /Format to *, the bar graph will appear automatically. See Table 2.3. Once you have it the way you like it, go ahead and save it.

Conclusion

Like everything else in the world of computers, to effectively use *The Spreadsheet* requires some forethought and practice. So, go ahead and try working with other variables. Even better, by adding and deleting rows, you can enter your own data set. Also, you might try to see how you can use the @IF function along with the logical @AND function to develop bivariate summary tables. Or, you may wish to wait for the next issue when we will explore how to examine bivariate relationships and create appropriate summary tables. See you then.

business

STRICTLY BUSINESS

Mike Regeimbal

PRODUCTIVITY has been characterized by many as working smarter rather than working harder. According to John Patton in his book "Patton's complete guide to productivity improvement," between 1889 and the end of World War II, productivity had increased in the U.S. 500%. Between then and the 1960's, average productivity increased at an annual rate of 3.1%. The 1970's saw annual productivity decline to 1.6% while inflation grew to 6% and hourly pay increased at a rate of 7.8%.

The last couple of years have brought about a renewed commitment by many of the leaders in this country at addressing the problem of productivity deterioration. The battle has been fought on many fronts. The realization that information flowing from the bottom up can be as beneficial as information flowing from the top down has resulted in wide spread acceptance of quality circles.

The need for increased productivity has added a new dimension of responsibility for those occupying the last step on the corporate ladder. The adage you cannot manage people into battle but must lead them has pushed corporate presidents into the limelight, resulting in leaders with "all the right stuff" being the focus of the American public.

The government has gotten into the picture through its efforts at deregulation. Every industry from farming to finance has been impacted by these efforts.

Surely national pride and competitiveness has had some impact. Many industries have utilized this premise as the basis for improving the quality of their product. The 12% annual productivity growth rate witnessed in Japan has provided a benchmark for setting our goals in this country. In the same vein, we as a country are again expounding that capitalism, with all its faults, is a far better economical alternative than a system which takes away your right to fail.

Still another aspect is the new awareness on the part of union and management personnel that we are no longer working in a national economy but a world economy. To be viable we must stay competitive. Obviously excessive wages at any level of the organization impacts competitiveness.

The urgency of the productivity problem has resulted in a win, win situation, not what I can do for you or you for me, but what can we do together. Working together we have made gains. In the month of September, 1983, the government reported that productivity increased by 5%. Fighting inflation and unemployment with productivity is working smarter rather than working harder.

One of the tools that has had a positive impact on productivity gains is the microcomputer. Just as the telephone, copier, and calculator have made their mark in the manager's environment, the microcomputer is being demanded as an indispensable tool. While uses for this new found tool are many and varied, the majority of managers are using it as a means of managing their information resources.

Large scale computing brought about the ability to complete highly repetitive tasks quickly and efficiently. The microcomputer has opened the doors for many to not only accomplish these same tasks, but go a step beyond, analyzing, stratifying, and dissecting that same information.

By knowing where information is, how to get at it promptly, and how to put it to work, the effective manager makes a significant contribution to productivity gains. Morton F. Metzler, in his book "Information: The Ultimate Management Resource," acknowledges that during the Industrial Age the emphasis was on worker productivity, today the emphasis is on management productivity. He states a manager's productivity is based on his or her ability to improve individual or organizational output. The microcomputer has helped managers evaluate information in four areas: accuracy, adequacy, reliability, and accessibility.

Even though information is not listed on the balance sheet, it is probably one of the most valuable assets of any company. The value of this asset depends to a large degree on the attitude of those in charge. There are those who would espouse limited information between departments, theorizing that disseminating such would compromise security, power, and foster misinterpretation. On the other hand, many organizations are aware of the synergistic effect of information dissemination, between departments, up and down the organization ladder. The value of information as an asset in these organizations is comparatively greater because of the synergistic effect and the risk of loss is spread out rather than concentrated with a few individuals. Obviously an investment in terms of training and equipment is required to fully develop the information asset. Some organizations are reluctant to make this investment but "if you think training is expensive try ignorance."

—◉→

business

Macroprint™ PLUS

"The most versatile Hi-Res / Lo-Res graphics & text interface card."

Four Boards in One!

Not only do you get the more powerful MACROPRINT mode (with many new features such as screen editing) but MACROPRINT PLUS also emulates commands (switch-selectable) of Pkaso,™ Grappler,™ Apple Parallel™ and Apple Dumpling™ boards.

Zooms Up To 15 Times Size!

Simple commands allow zooming with the same or different scale factors in either direction (vert/hor) for part or all of Hi-Res page; inversion, rotation, change of margins, side-by-side dumps and much more.

MACROPRINT PLUS interfaces to popular makes of Centronics compatible printers (switch-selectable) e.g. Apple Dot Matrix, Epson, C-Itoh, NEC, Gemini and Okidata. Use the same commands for all printer types when in MACROPRINT™ mode!

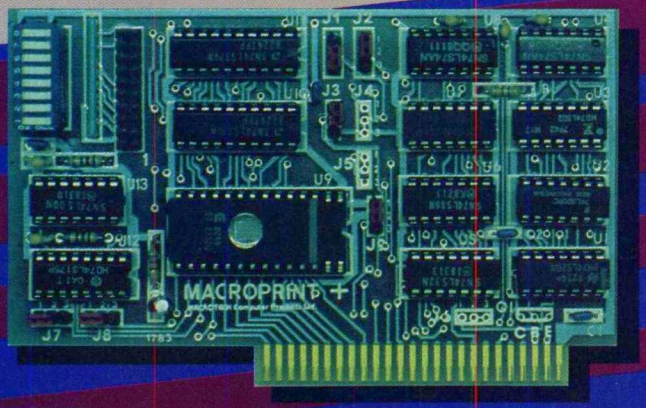
***Designed especially for* Apple II/II+/IIe,™ Franklin Ace™ & other Apple compatibles.**

Built-in Expansion Capability!

MACROTECH's unique design makes it the first printer interface card that can accommodate a larger EPROM (i.e. up to 256K program chip!) as made available. This means that new utilities/ features can be added as later options such as our famous MACRODISK™ and SPEEDISK™ memory utilities.

Suggested Retail price US$179. For more information contact your dealer or MACROTECH. Dealer and distributor inquiries most welcome.

Macroprint Plus, Macrodisk and Speedisk are TM's of Macrotech Computer Products Ltd., Pkaso TM of Interactive Structures Inc., Grappler TM of Orange Micro Inc., Apple Dumpling TM of Microtek Inc., Apple II/II+/IIe, Apple Parallel TM's of Apple Computer Inc., Franklin Ace TM of Franklin Computer Corporation.



Macrotech
COMPUTER PRODUCTS LTD.
Innovative Enhancement Technology
1370 Marine Drive, North Vancouver, Canada V7P 1T4 (604) 984-9305

Circle 11 on Reader Service Card

STRICTLY business

If information is a medium of increasing productivity and the microcomputer is the tool that will be used, to what functions then should the manager apply his efforts? Harold Koontz and Cyril O'Donnell in their book "Principles of Management: An Analysis of Managerial Functions," have identified five broad areas to which the manager must direct his efforts.

Planning is deciding in advance what to do, how to do it, and who is to do it. Planning bridges the gap from where your organization is to where you want it to go. It makes it possible for things to occur which would not otherwise happen.

Organizing is designing and maintaining groups of activities necessary to obtain objectives. Accordingly it must include the authority necessary to supervise it, and the provision for coordination horizontally and vertically in the enterprise structure.

Staffing is manning the organization structure through proper and effective selection, appraisal, and development of personnel to fill the roles designed into the organization.

Directing is leading subordinates toward effectively and efficiently attaining the objectives of the business enterprise.

Controlling is the measurement and correction of the performance of activities of subordinates in order to make sure that enterprise objectives are being accomplished.

In the five months that follow, we will explore each of these management functions in detail. The objective of these in-depth studies will be to identify how we, as managers, will be able to utilize the microcomputer, (our tool for information resource management) in each of these functions. Our focus will be directed toward methods of increasing not only managerial, but organizational productivity as well. So that we may gain a synergistic effect on these future articles, I invite each of you to share with the rest of the club your experiences using the microcomputer in various managerial functions.

BUSINESS

Software Review

EZ LEDGER is designed for the businessman using the cash (vs. accrual) system of accounting. Included in this category would be professionals such as doctors, lawyers, accountants, engineers, etc. Under the cash system of accounting, income is recorded when money is received and expense is recorded when money is dispersed. EZ LEDGER will keep track of expense items utilizing up to 99 different account titles. In addition, these expense accounts can be further classified as deductible and non-deductible, easing some of the burden at tax time.

This program offers holding files for both accounts receivable and accounts payable which automatically update the expense or income account when money is paid or received. These holding files carry their own year to date running totals. Addition or deletion of items in the expense, income, receivable, and payable accounts is fast and easy.

The EZ LEDGER program includes provisions for printing automatically numbered invoices. The invoice menu includes a recording date, salesman, customer purchase order number, carrier, terms, six lines for sold to and ship to, quantity, description, unit price with automatic extension, sales tax, and shipping and handling charges.

In addition no less than eight different reports can be generated at the touch of a key. Expenses can be formatted for year to date or monthly totals. Income can be displayed by type, month, or year to date recap, along with the traditional profit and loss statement. Year to date accounts payable and accounts receivable totals can also be reported.

The capability of deleting the constant query "Is this information correct Y/N" eliminates a lot of added effort on the part of the expert user, while providing the needed escape clause for the novice. This program will support both 80 and 132 column printers. The documentation is well formatted and easy to follow when and if you require it.

business

Macromem-3™

64-128K Memory Board

Adds 64 to 128K to your Apple II/II+/IIe.™ MACROMEM-3 is used automatically by programs like MAGICALC™ spreadsheet for extra file space and speed! MACROMEM-3 also operates as a 16K language card in an Apple II/II+ when installed in slot 0 for Pascal and other programs. MACROMEM-3 uses no cable!

MACROTECH Computer Products Ltd. also offers DISKULATOR™ (64–512K) for those requiring more than 128K. For smaller requirements, MACROMEM-1 (16K) and MACROMEM-2 (32K) are also available. Contact your dealer or MACROTECH for more information.

Dealer and distributor inquiries most welcome.

Macromem, Macrodisk, Speedisk, Macrospeed, and Diskulator are TM's of Macrotech Computer Products Ltd. Apple II/II+/IIe are TM's of Apple Computer Inc. Magicalc is a TM of Artsci Inc.

Accompanying Software includes these powerful utility programs:

1. MACRODISK™
–Disk Emulation

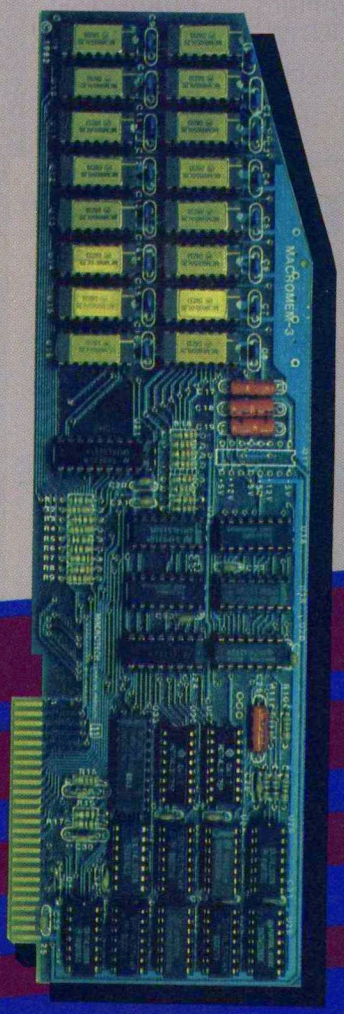
Enables the board to emulate a disk drive for high speed access in either DOS 3.3, Pascal, or CP/M.

2. SPEEDISK™
–Disk Access/Backup

No more need to load and save files to/from memory board as with RAM-Disk programs. With SPEEDISK, disk access speed nears that of RAM board access time plus all new data is automatically backed up on disk!

3. MACROSPEED™
–Disk Duplication

Use the memory board to speed up disk duplication dramatically!



Macrotech
COMPUTER PRODUCTS LTD.
Innovative Enhancement Technology
1370 Marine Drive, North Vancouver, Canada V7P 1T4 (604) 984-9305

We Are Not PIRATES

But We Can Show You How To

UNLOCK LOCKED-UP SOFTWARE

with our complete program listings and step-by-step instructions

YOU CAN, TOO!

CORE

IN-DEPTH COVERAGE
of
Special Interest Subjects

- GRAPHICS
- DATA BASES
- UTILITIES
- GAMES

For Hobbyists and Advanced Computists

Published Quarterly

hardcore

HARD FACTS
on
APPLE SOFTWARE

An Unprecedented and Controversial disclosure of closely-held

COMPUTER SECRETS

How to: Back-Up Disks
Customize
Commercial Programs
Undo Copy-Protection & More!

Published 8 times yearly

Make checks and money orders payable to:
CORE/HARDCORE Computist
P.O. Box 44549H Tacoma, WA 98444

Annual Subscription Rate:
$25.00 for 8 issues of HARDCORE COMPUTIST and 4 issues of CORE.
Canada: **$34.00** Mexico: **$37.00**
Overseas, foreign countries: **$47.00.**

Name:
Address:
City: State:
Country: Zip:

Payment must accompany order.
U.S. funds only. No credit cards.

Circle 40 on Reader Service Card

COMMUNICATION PROBLEMS?

PERSON-TO-PERSON™
PROVIDES THE
INTEGRATED SOLUTION

Data, voice and mail, PTP 1.1 handles them all.

PTP keeps a single comprehensive phone, address and memo database. With this file, it prints, dials and runs computer communications. Perfect for the home. A productivity tool for business.

- Dials voice calls, can use non-Bell long-distance, send tone-dialed data
- Displays any address in 2 seconds, revise memos during calls
- Print an envelope, filing card, mailing list or merge-print form letters
- Up or download data files or electronic mail
- Auto log-on or prompted auto answering with auto up or downloading
- Alternate voice and data

PTP has interactive prompting and is totally menu-driven. It's convenient, versatile, fast and friendly.

Person-to-Person. The integrated communications database for the Apple II and //e from Trutec Software.™ Only $69.95.

Ask your dealer for a demo. Or order direct by phone or mail. Credit card orders accepted by phone. By mail send check or money order. Add $3.00/shipping. California residents add 6% sales tax. Dealer inquiry invited.

Minimal system: 48K, 1 disk drive. Optional: modem,* printer, 16K card, 2nd disk drive, 80 column card, //e features. Capacity (64K) about 1400 listings per file. Also suitable for general filing.

*Modems supported: Hayes Micromodem II or IIe, Smartmodem 300 or 1200 (with Apple Super Serial Interface), Novation Apple-Cat II and compatibles of these.

Apple is a registered trademark of Apple Computer. Micromodem II and Smartmodem are registered trademarks of Hayes Microcomputer Products. Apple-Cat II is a trademark of Novation.

TRUTEC™ SOFTWARE

1700 Solano, Berkeley, CA 94707
Orders: 800-621-3744 (In CA 415-525-4901)

Circle 28 on Reader Service Card

STRICTLY

I enjoyed using this program and feel it would be very useful to those on the cash basis of accounting. EZ LEDGER is written by John Bridges and distributed by Highlands Computers, 14422 S.E. 132nd St., Renton, Washington 98056, area code (206) 228-6691. Price is $60.00.

Book Review

This month's selection is "Getting Down To Business With Your Microcomputer" by James A. Gupton, Jr. This book is targeted toward the person who does not have a microcomputer and is contemplating purchasing one for use in the home or office.

The first 80 pages of this book focus on the inner workings and peripherals for microcomputers. The discussion includes a review of the power supply, semiconductor memory (both RAM and ROM), busses, and interfaces. Additionally, mediums of input and output are explored in detail. The next 80 pages present a thorough discussion of microcomputer systems, their capabilities, and characteristics.

The last 40 odd pages cover potential uses of the microcomputer. Uses in the home include security and climate control applications. For the business person, the book considers applications for general ledger, accounts payable, accounts receivable, payroll, and inventory management. In addition, a brief discussion of the potential use in the legal or medical office is given. The glossary of terms and the appendix which explains decimal, binary, octal, and hexidecimal numbering systems can be very helpful to the novice.

I would recommend this book to those people who do not have a microcomputer and will be purchasing one in the near future. In my opinion this book does not present the level of detail required for those people who already own a computer and are looking to upgrade to another unit.

Next month we will be discussing using the microcomputer for the planning function. I hope all of you had a nice holiday season and that Santa left you a new Apple under the Christmas tree.

business

FLYING HIGH . . . WITH E-Z PILOT

Earl Keyser

WELCOME back to Flying High, the column which will tell you everything you wanted to know, but were afraid to ask, about E-Z PILOT, *Call -A.P.P.L.E.'s* version of the language PILOT. This month we will discuss two things: moving about within your lesson and concluding your lesson.

Actually, let's take last things first. That is, let's discuss how to END a lesson. We'll embellish our last program, the one about the largest state.

As of last time, we had —

```
T: Which state is the largest
A:
M: ALASKA,ALASCA,ALLASKA
TY: Right.
M: TEXAS
TY: No, only close.
```

Now, you and I know that the end of the lesson is the last line. But, E-Z PILOT needs to be told. So, we need to add an E: or END statement at the conclusion of the lesson. Like this —

```
T: Which state is the largest
A:
M: ALASKA,ALASCA,ALLASKA
TY: Right.
M: TEXAS
TY: No, only close.
E:
```

Now, we have properly concluded the lesson.

But, let's go a bit further and test the student's knowledge of geography a bit more and, in addition, give him or her one more shot at the question. We'll enter the lesson again a bit differently and add a few lines.

Boot up your E-Z PILOT disk. When you reach the menu of demonstration lessons, type AUTHOR, instead of the number for a lesson. You will quickly be sent to AUTHOR, the part of E-Z PILOT with which you write lessons.

The 12 options in the top box are all the things you can do in AUTHOR. They will allow you to create, edit and manage your lessons. For the moment, type WR. That stands for WRITE. You need only type the first TWO letters of an option. Answer the question "Are you sure?" with YES. The question is there to ensure that you don't erase a good lesson which you had not saved.

Now, you'll be sent to a screen where you may enter the lines of your lesson. Each line is numbered for reference in editing. The numbers are not part of the lesson itself and serve *no* purpose except for editing your lesson. So, begin typing in this lesson.

```
T: Which state is the largest
A:
M: ALASKA,ALASCA,ALLASKA
TY: Right.
JY: *H
M: TEXAS
TY: No, only close.
T:
T: And is the furthest state north
A:
M: ALASKA,ALASCA,ALLASKA
TY: Excellent guess. Alaska is
TY: our largest state.
TN: Oh, sorry. It's Alaska.
*H:
T:
T: Question number two is easier.
T:
T: Which state is furthest south
A:
M: HAWAII,HAWII,HAWYE,HUW
TY: Great. First shot. Hawaii is
right!
JY: *E
M: TEXAS
TY: Still like the Lone Star State,
eh?
TY: Well, you're also still wrong.
T: It's Hawaii, the Aloha State.
*E:
E:
```

Now, this is a bit more complicated. Here's what's happening.

We have seen lines 1 through 7 in the previous example. Except for line 5. Line 5 is the way one moves around within a lesson. To move around, you use a JUMP statement. JUMP statements, like kangaroos and frogs, have to JUMP somewhere. Lacking an outback or a lily pad, E-Z PILOT has JUMPs which jump to lines known as labels. In E-Z PILOT, a label is stored on a line alone. Some PILOTs allow labels to precede other lines, but not here.

When E-Z PILOT encounters a J:, it looks for the label which is to the right of the colon. That label must consist of an asterisk (a star) and ONE letter or digit. Sadly, you can't have labels like *WARS and *TREK and *BRIGHT. Maybe in version #S3.14159265KY. Or, maybe that's just pi(e) in the sky. . .

Wherever that label is in the lesson, E-Z PILOT will JUMP to it and continue the lesson from there. In our example, we have modified the J: with a Y. This would mean that, if the result of the previous match were true, then E-Z PILOT would jump to the label, *H, which stands for Hawaii, the subject of our second question.

If a match is not found, a jump will not be made and the student would be asked a second question. If he/she gets it right this time, they would be told so in lines 12 and 13. If they really don't know, we would give them the answer.

Line 15, the *H, begins our second question. The format of the question is similar to the previous one, except that we jump to a label immediately preceding the E: statement if the student gets the answer right. Note that the M:TEXAS resets the match flag and the succeeding Y and N modified statements are affected only by the result of the LATEST match.

So, we've now covered E:, for END and J:, for MATCH. These are the real heart of any version of PILOT. With T, A, M, J and E, you can create many, excellent lessons. Actually, even very young children (7 and up) have used PILOT successfully to make lessons and pictures.

(continued on page 51)

education

APPLE II Outperforms IBM PC with 'Supermini' Virtual Memory Technology

If you need to get SERIOUS, POWERFUL, PROFESSIONAL results from your II+/e, for Business or Engineering applications, nothing comes close to MegaTASK PLUS! Modelled on and developed from minicomputer-based commercial timesharing software available to large corporations for thousands of dollars annually, MegaTASK PLUS is primarily designed for the APPLE II owner who demands quick, effective analytical solutions to business problems—and immediate RELIEF from the frustrations and limitations of using hundreds of dollars of mutually-incompatible 'canned' software packages, or WORSE, programming in Applesoft, assembly or Apple DOS!

MegaTASK PLUS is a combined, multi-product hardware-and-software package that provides an instant **Virtual Memory** expansion, an Applesoft-compatible **SuperBASIC**, and a series of **Business Management** packages that greatly enhance the power and value of the APPLE II in both **systems programming** and **business applications** areas. MegaTASK and MegaDOS employ multitasking, virtual memory optimization and virtual storage technology, which, until now, were limited only to mainframes and the 'super' minicomputers. The SMARTCHIP™, an Intel 8748 Single-Chip Plug-In Microcomputer, provides sufficient processing power to enable your APPLE II to outperform both the IBM PC and Apple's new MacIntosh computer. Yet, all you need to begin is an APPLE II series computer plus one diskette drive. And generally, no modification to your existing Applesoft programs, assemblies and DOS 3.3 files is necessary.

MegaTASK and MegaDOS accordingly generate an enormous enhancement to the power available to the APPLE II owner, specifically:

- Multitasking System processing techniques. (As many as 16 PROGRAMS or TASKS can execute at one time).
- The MegaDOS VS/OS (Virtual Memory Operating System) can place within reach of your system MEGABYTES of Virtual Memory. Largest program size: 8 Megabytes. Largest ISAM file: 451 Megabytes. Largest transparent core array: 75 million FP numbers.
- 100-400 new Mainframe Basic commands bring machine-language speeds to interpretive Basic. Powerful instructions comprise extensions to Applesoft that include many of the most useful commands found in FORTRAN, COBOL, PL/1, ADA, PASCAL and LOGO.
- Enormously simplified ease of programming—modular, insulated routines, localized variables, ability to pass variable names. Dependence on line numbers **totally removed.**
- Substantially increased speeds: program execution (3x-10x), file access (6x-10x), alpha data transfer rate (2x-6x), and numeric transfer (>20x).

On the applications side, the 'PLUS' portion of MegaTASK PLUS includes:

- Hardcopy and screen Business Graphics, Calc, Database Management and Word Processing Applications Packages—plus Custom Modifiability.
- Editing Systems & Format-Related Commands
- Communications/Corporate Data Capture programs
- Text Manipulators and better-than-TSO Program Editors
- Database Management Systems capable of maintaining and sorting Supermini-Size Datafiles. Maximum single transparent file: 5.5 million records (if 80-chrs ea). File access time independent of file size.
- Report Generators and Strategic Business Analysis programs—similar to those used by most of America's Fortune 500 Corporations.
- The only system capable of taking full, transparent advantage of one or more Corvus's, Profile's, Ramdisks, 5-Megabyte replaceable hard-disk cartridges and the coming Laser Disk units, bringing the APPLE II up to minicomputer performance.

The MegaTASK PLUS product package includes:

- Manuals for MegaTASK, MegaDOS and MEGABASE II in sparkling white binders
- 1 SMARTCHIP™ Memory Management Microprocessor
- Six diskettes PACKED with applications programs, including Mega GRAPH, Mega EDIT, Mega TEXT and several Report Generator examples.
- A One-year Replacement Warranty on all products.
- A 60-day Financial Risk-Free Return Policy. If you are not completely satisfied you can return the product in good condition with proof-of-purchase for a complete refund.
- A 14-hour-per-day telephone Hotline to support you in your use of the entire MegaTASK PLUS series of software packages.

Because of the much larger amount of memory that is available to the Apple's microprocessor through Virtual Memory, the increased processing speed, and the integration of major applications packages PLUS custom modifiability by users who know Basic, MegaTASK PLUS may well surpass or replace most of the worthwhile APPLE II program packages currently available on the market. With the introduction of this product, Quantum Leap has made a commitment to be the technology leader in the area of business software systems for personal computers. This will be achieved through Quantum Leap Technologies' continuing Applications Software Package Series—quarterly diskettes and manuals mailed free to purchasers of MegaTASK PLUS for one year from date-of-purchase.

Quantum Leap software systems are designed to give you, the user, powerful professional and business tools with which to get the day's work done faster, more pleasantly, and more professionally. We live in an exciting age for personal computing. **Multitasking and Virtual Memory** promises to make any computer substantially more powerful than is possible with currently limited memory (Byte Magazine Editorial, March 1983). Only MegaTASK PLUS offers you these technologies. PLUS, standard Applications Packages to capitalize on them—all available for your APPLE II!

$495.

QUANTUM LEAP
TECHNOLOGIES

ONE CORPORATE CENTER • BOX 6440
ARLINGTON, VIRGINIA 22206 • (703) 548-5308
Q.L. HOTLINE: (703) 683-0308

Also distributed on the West Coast by:
Data Link, 1225 Jefferson Davis Hwy., Arlington, VA 22202. (703) 486-5701

Apple is a trademark of Apple Computer, Inc. IBM is a trademark of International Business Machines

Circle 22 on Reader Service Card

A Call —A.P.P.L.E. BOOK REVIEW:

MIND STORMS

Eric Flescher/reviewer

Mindstorms: Children, Computers and Powerful Ideas
by Seymour Papert
Basic Books Inc., Publishers
New York, N.Y.
1980

THE field of computer science and computer science education has engulfed many parts of our daily lives. Accompanying this, has been a proliferating flood of books. This deluge of books for the most part, cover many aspects of the effects or uses of the high technology. For the most part, few of the books deal with the most important nature of these devices. How computers can affect the way people think and learn and the true difference they can make. One brilliant text does: Mindstorms.

The author, Seymour Papert, has written a most valuable book for anyone wishing to utilize computers to their full potential. More importantly he has conceived, through the use of his Logo language and his philosophies, an exciting vision for education of the future. He elaborates and draws you into the significance of his ideas through theoretical and practical information. Most importantly, he crystallizes the ideas through his unique view: as an educator, a computer scientist blended with a psychologist point of view.

Computers are now more commonplace in the schools. However, for the most part, equipment is preceding the knowledge, training and materials needed to utilize the devices to their fullest. More importantly, what is missing is knowledge concerning the computers and how they may help the thinking and learning processes. These are the topics in which Dr. Papert and his book excel, possibly better then any other publication on the market today.

In the introduction, Dr. Papert relates his educational philosophies. These grew from his associations with Jean Piaget, whose philosophies and works helped understand the intellectual learning processes of children. Dr. Papert emphasized that certain learning does take place early. He also says that certain obstacles tend to restrict the learning process (p.7). In the following eight chapters, Dr. Papert talks about such obstacles. He expounds upon his belief that computers can:

1) Be carriers of powerful ideas.
2) Can help in the challenging about who can understand what and what age.
3) Assist us in a better understanding of how learning can work, not blocked, by cutting across traditional and artificial boundaries (p.4).

In the first chapter, "Computers and Computer Cultures," Papert states that the devices can help children explore how they themselves can think (p.19), can help develop deliberate styles of thinking (p.27), gives an opportunity to guide intellectual pursuits while drawing from the surroundings (p.32).

"Mathophobia: The fear of learning," the second chapter, centers on math as a subject, which taught alone, stifles many. The author stresses that his "turtle geometry" allows children and adults to design personally meaningful experiences that are part of the natural learning process. This is expounded upon in the third chapter, "Turtle geometry," in which the author stresses that "associative (synotic) learning" (p.63), utilizes problem solving and other strategies naturally. A dialogue between two children (p.77–93) illustrates how learning can become more dynamic and how this comes about through the use of the Logo language.

The fourth chapter, "Languages for Computers and for People," keys on the fact that children can be taught and can learn from mistakes. The machines and appropriate methods can help to show that sharing of problems and experiences and collaboration can help learning (p.114). In the fifth chapter, "Microworlds: Incubators for knowledge," expands upon this. The author stresses that computer based interactive learning can actively use prerequisite knowledge and apply it towards a personally defined end.

In "Powerful ideas in Mind-Size Bites," (sixth chapter), Papert states that the computers can help "in getting to know (qualitative)" versus "what to know (quantitative)." The devices can help in exercising intuitive knowledge. "Logo's Roots: Piaget and AI," the next chapter, enhances the idea that an innovative language/use of computers can aid in gaining a deeper insight into what can be learned.

The eighth chapter, "Images of the Learning Society," stresses that it is an evolution of ideas, not just technology (p.186). It can help in the better understanding of many subjects and the process of learning itself.

These last statements are the most important. As you read through the book, you can gain a understanding of the interactive role of the computer. One can gain a better appreciation of this by trying to learn how to use computers, first hand, via a course, books or self teaching. Wishing to use the devices as a means to promote innovative education, I found this book most valuable in developing my personal philosophy. After reading it, I came away with a better understanding of how to reach my goals. I gained a clearer understanding of how computers could foster creativity, individualized education, and multidisciplinary learning.

(continued on page 51)

education

A Call —A.P.P.L.E. SOFTWARE REVIEW:

FACEMAKER

Garry Margolis/reviewer

Facemaker *$39.95*
Published by Spinnaker Software Corp.
215 First Street
Cambridge, MA 02142
Copy Protected

MUCH noise has been made about Logo and turtle graphics as a tool for introducing children to concepts of programming. However, Spinnaker Software, which has its own turtle graphics program called *Delta Drawing,* has an even better introduction to computer programming for young children.

Facemaker is a superb teaching program for 4–8 year olds disguised as a whimsical game. Written by Clark Quinn and Margaret Weinstein of DesignWare, it has captivated our resident 5½ year old tester as well as every adult in our vicinity who has been exposed to it. Although "Mr. Potato Head," the old children's game, is somewhat similar, Facemaker far outstrips it with unique and valuable features.

Upon booting, a brief menu of choices is presented (sound on/off, white/black background, help). After these are answered, the child presses "1" and a blank oval face is put on the right side of the screen. A menu denoting eyes, ears, nose, mouth and hair is on the left — the child advances sequentially through these features using the SPACE bar and chooses one group by pressing RETURN. After this choice is made, a sub-menu appears with eight delightful variations on the chosen feature. For example, among the choices for ears are pointed, Spock-like ones and miniature ear trumpets. Again, the child uses the SPACE bar to advance through these features — a press of the RETURN key places that feature in its place on the blank face and the original menu returns. Unlike "Mr. Potato Head," the features always are placed in their proper position on the head.

The child builds a face up, feature by feature, and can change any one at will. There are 32,768 combinations available, and the artwork is cleverly designed so that each feature combines well with every other one.

When the child is satisfied with the face that he or she has assembled, a press of the numeral "2" goes to the next module. This section is the most ingenious and valuable part of Facemaker. The child can make the face wink, frown, smile, cry, stick out its tongue and wiggle its ears by pressing "W", "F", "S", "C", "T" and "W", respectively. Each movement is accompanied by an appropriate sound effect — as noted above, the sound can be turned off if desired. Even better: the child can type a sequence of up to 14 of these commands and then, by pressing RETURN, cause the face to perform these actions in that sequence. The child is actually writing and executing a program by issuing commands for immediate or deferred execution!

Although Logo is the "hot" method of introducing children to programming, as stated above, Facemaker's animation section is an even more useful first step toward teaching programming concepts, because the child does not have to start with a blank slate and draw a picture — the picture is assembled using menus and then programmed with single-key commands or sequences of keystrokes. Logo is a logical next step after Facemaker, either by itself or in the *Delta Drawing* variant, but it need not be the child's first exposure. Our resident tester will be introduced to turtle graphics soon and will probably have little trouble with it because of his experience with programming Facemaker.

But there is still more to Facemaker. Pressing "3" switches to a "Simon Says" game, where the assembled face executes a progressively longer sequence of actions and the child is expected to key in the corresponding letters in correct sequence. At the beginning, where the sequence consists of only two actions, it is relatively easy for most children to respond correctly, and the face responds with a wink and a smile, while "Charge!" is quickly played (in tune!). As the sequences increase in length and difficulty, children (and adults) will eventually reach their limits, and when an error is made, a quick "Nyah nyah-nyah nyah nyah" is played and the face sticks out its tongue with a razzing sound, after which a new sequence begins. However, the response to an error is so interesting that some children might prefer erroneous input.

Our resident tester, as stated above, is absolutely enthralled by Facemaker. He caught on to the SPACE and RETURN method of menu selection in five minutes and the programming system in ten. As an experiment, he was left alone with the computer after his first fifteen minutes with Facemaker — he spent an astonishing one hour and fifteen minutes by himself, chortling with glee as he put together one silly face after another and got each to wiggle its ears and stick out its tongue, occasionally calling me in to share a particularly funny combination.

Facemaker is, regrettably, copy protected. There is a 30 day dealer exchange privilege for non-booting diskettes; after that time, damaged diskettes will be replaced by Spinnaker for $5.00. The graphics are excellent on a monochrome screen. RESET reboots the program, and it is otherwise crash-proof, as any input which it does not expect is ignored.

education

MIND

(Continued from page 49)

It's not a "cookbook," a book that will teach you how to program or learn a computer language. The book's strengths lie in its ability to relate the powerful nature of computers as a means to enhance, not replace, current educational practices. Dr. Papert in this book, has laid out a "map" for us to follow. His effort is not one of selling his language and its virtues. It is a sincere effort to broaden our understanding of the machines and the dynamic, interactive role that the high technology can play in the educational process.

This book is not as "cut and dried" as other texts. The so called, innovative learning and the use of computers takes different methods, ideas, and techniques. It is after all, a different, not necessarily new way of thinking. If you gain an insight into some of these new ideas and begin to think of the possible uses in the classroom or at home, then you have learned some of the powerful ideas Papert expounds upon in Mindstorms.

FACEMAKER

(Continued from page 50)

Spinnaker has also released this program for several computers besides Apple, but I have not seen these and consequently cannot comment on how well the other versions work. However, the program is written in MicroMotion Forth-79 and, thus, may be reasonably transportable.

In sum, Facemaker is a brilliantly conceived and executed program. Its outstanding educational content is matched by superb entertainment value. Highly recommended!

FLYING

(Continued from page 47)

Here's one to try:

```
T: Hello there, person.
T:
T: I am going to make a picture.
T:
*P:
T:    *
T:   ***
T:  *****
T: *******
T:  *****
T:   ***
T:    *
T:
J: *P
E:
```

A hint. Be prepared to hit CONTROL and the letter C.

Next time, we'll get into R, C and the mysterious U. We'll also deal a little bit with the fun stuff in E-Z PILOT like B-I-G letters and noises.

Until then, wave your wings as you fly by.

education

ACCOUNTS RECEIVABLE -ONLY-

$75

-A SIMPLE, RELIABLE AND INEXPENSIVE SOLUTION TO KEEPING TRACK OF YOUR ACCOUNTS RECEIVABLE.

ACCOUNTS RECEIVABLE — ONLY — WAS DESIGNED BY A PROFESSIONAL BUSINESS/ACCOUNTING TEAM TO GIVE SMALL BUSINESSES THE ADVANTAGES OF IMMEDIATE ACCESS TO AND CONTROL OF THEIR ACCOUNTS RECEIVABLE WITH THE SPEED AND ACCURACY OF MUCH LARGER AND MORE EXPENSIVE SYSTEMS

SYSTEM FEATURES

- POSTS YOUR INVOICES TO A -BALANCE FORWARD- AGED ACCOUNTS RECEIVABLE SYSTEM.
- NO FORMS NEEDED - PRINTS PROFESSIONAL-LOOKING FORMS ON PLAIN PAPER (SAVES TIME AND MONEY).
- PRINTS STATEMENTS, POSTING JOURNALS WITH MTD TOTALS, TRIAL BALANCE AND AGING REPORT.
- PROVIDES YOU WITH THE EXACT STATUS OF YOUR ACCOUNTS RECEIVABLE AT ANY TIME.
- HANDLES 300 ACCOUNTS PER DISK SET.
- SAVES HOURS OF TIME.

System is Fully Copyable for Back-up Convenience. Requires Apple //e* or Apple//+, Two Disk Drives, Printer.

● PROFESSIONAL TELEPHONE SUPPORT ●

CALL OR WRITE TO ORDER

ACCOUNTS RECEIVABLE-ONLY- $75
(SYSTEM AND MANUAL)

Contact Us for Information and Sample Print-outs.

ALPINE COMPUTER SYSTEMS

ALPINE COMPUTER SYSTEMS
P. O. Box 128
ALPINE, TEXAS 79831
(915) 837-2501

*Apple is a Registered Trademark of Apple Computer, Inc.

Circle 2 on Reader Service Card


```
]LIST

5  REM

   SO WHAT DID YOU EXPECT

MARK ALLEN : CALL -A.P.P.L.E.

10  HOME : VTAB 12: PRINT "WHAT'S
    YOUR NAME?"
20  FOR I = 1 TO 20: GOSUB 110: GET
    Z$
40  IF Z$ =  CHR$ (13) THEN 100
50  IF Z$ =  CHR$ (32) OR Z$ =  CHR$
    (21) THEN 80
60 Z =  INT ( RND (1) * 90): IF Z
    < 65 THEN 60
70  PRINT  CHR$ (Z): GOTO 90
80  PRINT  CHR$ (32)
90  NEXT I
100  VTAB 18: HTAB 7: PRINT "SO W
    HAT DID YOU EXPECT?": FOR I =
    1 TO 3000: NEXT : END
110  VTAB 12: HTAB (18 + I): RETURN
```

APPLE MECHANIC

HI-RES SHAPE EDITOR / TYPE FONT DISK
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #5.

SHAPE EDITOR: Keyboard-draw hi-res shapes for animation in your Applesoft programs. Access & create **proportionally-spaced hi-res Typefaces** with each character re-definable as you want. Six fonts are included on the disk. Excellent LISTable Applesoft demos show you how to animate graphics and create professional-looking Charts and Graphs.

BYTE-ZAP: Rewrite any byte on a disk for repair or alteration. Load entire sectors on the screen for inspection. **Hex/Dec/Ascii displays** and input. Educational experiments included for making trick file names, restoring deleted files, changing DOS, etc.

MORE: Useful music, text and hi-res tricks for your programs. Clear educational documentation.

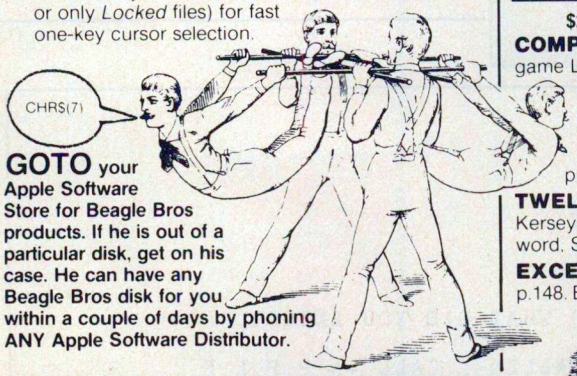
APPLE MECHANIC TYPEFACES

by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

26 NEW FONTS for use with Apple Mechanic programs. Many different sizes and typestyles, both ordinary and *Artistic*. Every character—from A to Z to "★" to "□"—of every typeface—from "Ace" to "Zooloo"—is re-definable to suit your needs. All typefaces are **proportionally spaced** for a more professional appearance. People *do* notice the difference!

BEAGLE MENU: Display only the file names you want from your disks (for example, only *Applesoft* or only *Locked* files) for fast one-key cursor selection.



GOTO your Apple Software Store for Beagle Bros products. If he is out of a particular disk, get on his case. He can have any Beagle Bros disk for you within a couple of days by phoning ANY Apple Software Distributor.

RUSH the following disks by First Class Mail—

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

Visa/MasterCard or COD, call **TOLL-FREE**

Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S.Check, Money-Order or Visa MC# to **BEAGLE BROS**, 8th Floor, 4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping. Any-Size Order. Overseas add $4.00. COD add $3.00. California add 6% sales tax.
ALL ORDERS SHIPPED IMMEDIATELY.

DOS BOSS

DISK COMMAND EDITOR
by BERT KERSEY and JACK CASSIDY

$24.00: Includes Peeks/Pokes Chart & Tip Book #2.

RENAME DOS COMMANDS & Error Messages—"Catalog" can be "Cat"; "Syntax Error" can be "Oops" or almost *anything* you want it to be.

PROTECT YOUR PROGRAMS. An unauthorized Save-attempt can produce a "Not Copyable" message, or *any* message you want. Also easy List-Prevention and other useful Apple tips and tricks. Plus one-key program-execution from catalog.

CUSTOMIZE DOS. Change the catalog Disk Volume heading to your message or title. Omit or alter catalog file codes. Fascinating documentation, tips and educational Apple experiments.

ANYONE USING YOUR DISKS (booted or not) will be using DOS the way YOU designed it.

10 LIST: LIST: LIST: FOR ZZ=PEEK(175)+PEEK (176)*256+36 TO 3072: POKE ZZ,216: NEXT 20 FOR XXX=1 TO 2: POKE-16299,0: POKE -16300,0: XXX=1: NEXT: REM Experiment with different length variable names.

BEAGLE BAG

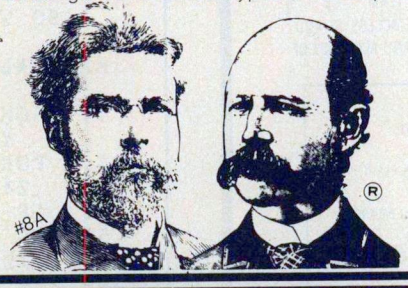
12 APPLE GAMES ON ONE DISK
by BERT KERSEY

$29.50: Includes Peeks & Pokes Chart

COMPARE BEAGLE BAG with any single-game Locked-Up disk on the market today. All 12 games are a blast, the price is a bargain, the instructions are crystal clear, *and* the disk is COPYABLE. You can even *change* the programs or list them to learn programming tricks by seeing how they work.

TWELVE GAMES from the Applesoft Ace, Bert Kersey— TextTrain, Wowzo, Magic Pack, Buzzword, Slippery Digits, and many many more...

EXCELLENT REVIEWS—See Jan-83 *Softalk*, p.148. Beagle Menu too: see Typefaces description.



Beagle Bros
Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.* (Don't Settle for Less!)

*DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who

NEW! SILICON SALAD

INCLUDING TIP DISK #2
by BERT KERSEY and MARK SIMONSEN

$24.95: Includes Peeks/Pokes AND Commands Charts

MANY MINI-UTILITIES: Disk Scanner finds bad disk sectors, **Key-Clicker** adds subtle sound as you type, **DOS-Killer** adds two tracks of space to your disks, **2-Track Cat** allows up to 210 file names per disk, **Program Splitter** makes room for hi-res pix with large Applesoft programs, **Text Imprinter** transfers text to the hi-res screen, **Onerr Tell Me** prints the appropriate error message but continues program execution, **Text Screen Formatter** converts text layouts into Print statements... plus much more Apple wizardry from the boys at Beagle Bros.

MORE TIPS ON DISK: Including fantastic programming tricks from Beagle Bros Tip Books 5, 6 and 7, plus programs from Tips/Tricks Chart #1.

TWO-LINERS TOO: From our customers around the world—and elsewhere. Little mind-blowers that will teach your old Apple some new tricks!

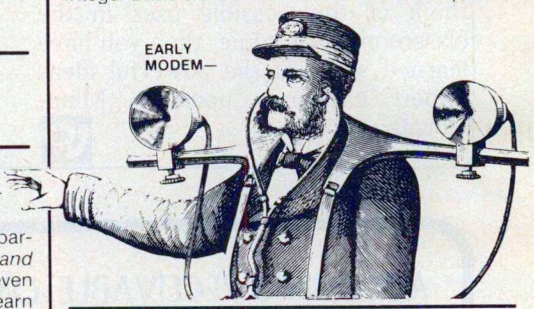
TIP DISK #1

100 TIP BOOK TIPS ON DISK
by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

100 LISTABLE PROGRAMS from Beagle Bros Tip Books 1-4. Make your Apple do things it's never done! All 100 programs are LISTable and changeable for Apple experimentation.

COMMAND CHART INCLUDED: Free with each Tip Disk; an 11 x 17 poster of all Applesoft, Integer Basic & DOS Commands with Descriptions!



FLEX TYPE

(FORMERLY "FLEX TEXT")
VARIABLE-WIDTH HI-RES TEXT UTILITY
by MARK SIMONSEN

$29.50: Includes Peeks & Pokes Chart

PRINT VARIABLE-WIDTH TEXT on both hi-res screens with normal Applesoft commands (including HTAB 1-70). Normal, expanded & compressed text with no extra hardware. (70-column text requires a monochrome monitor, not a tv).

ADD GRAPHICS TO TEXT or add Text to hi-res graphics. Run your existing Applesoft programs under Flex Type control. Fast, easy to use, and Compatible with GPLE and Double-Take.

DOS TOOL KIT* font compatibility, or use the supplied Flex Type typefaces. Select up to 9 fonts with control-key commands. A text character editor lets you redesign any Apple text character.

FRAME-UP

FAST APPLE DISPLAY UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

PROFESSIONAL PRESENTATIONS: Turn your existing Hi-Res, Lo-Res and Text frames into attractive Apple "slide shows". *FAST* hi-res loads in 2 -seconds! Paddle or Keyboard-advance frames.

UNATTENDED SHOWS are optional, with each picture arranged and pre-programmed to display on the screen from 1 to 99 seconds. Custom **Text Screen Editor** lets you create black-and-white text "slides" and add type "live" from the keyboard during shows. Mail copies of presentations on disk to your friends and associates (or home to Mom!).

NEW!

GPLE

GLOBAL PROGRAM LINE EDITOR
by NEIL KONZEN

$49.95: Includes Peeks/Pokes Chart & Tip Book #7.

A CLASSIC APPLE PROGRAM EDITOR GPLE lets you edit Applesoft program lines *FAST* without awkward cursor-tracing and "escape editing".

INSERT & DELETE: GPLE works like a word processor for Applesoft program lines. You make changes instantly by jumping the cursor to the change point and inserting or deleting text. No need to trace to the end of a line before hitting Return.

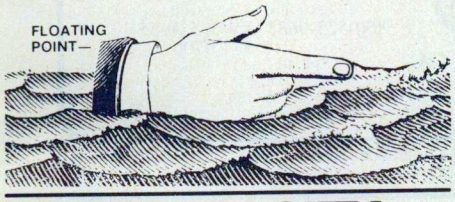
GLOBAL SEARCH & REPLACE: Find any word or variable in your programs, *FAST*. For example, find all lines containing a GOSUB, or edit or delete all lines with REM statements, or all occurrences of any variable. **Replace any variable**, word or character with any other. For example, change all X's to ABC's, or all "Horse" strings to "Cow".

80-COLUMN COMPATIBILITY: All edit & global features support **Apple IIe 80-column cards** and most 80-column cards on *any* Apple IIe, II+ or II.

DEFINABLE ESC FUNCTIONS: Define ESC plus any key to perform any task. For example, **ESC-1** can catalog drive 1, **ESC-L** can do a "HOME: LIST", **ESC-N** could type an entire subroutine... *Anything* you want, *whenever* you want.

GPLE DOS MOVER: Move DOS and GPLE to Language Card (or IIe upper 16K) for an EXTRA **10,000 Bytes** (10K) of programmable memory.

Plus APPLE TIP BOOK #7: Learn more about your Apple! Includes all new GPLE tips and tricks.



UTILITY CITY

21 PROGRAMMING UTILITIES
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #3

LIST FORMATTER prints each program statement on a new line. Loops indented with printer Page Breaks. A great Applesoft program de-bugger.

MULTI-COLUMN CATALOGS, with or without sector and file codes. Organize your disk library.

INVISIBLE and trick catalog file names. Invisible functioning commands in Applesoft programs too.

MUCH MORE: 21 utilities, including auto-post Run-number & Date in programs, alphabetize/store info on disk, convert dec to hex or Int to FP, protect and append programs, dump text to printer...

LEARN PROGRAMMING: List-able programs and informative documentation. Includes Tip Book #3. Hours of good reading & Applesoft experiments.

ALPHA PLOT

HI-RES GRAPHICS/TEXT UTILITY
by BERT KERSEY and JACK CASSIDY

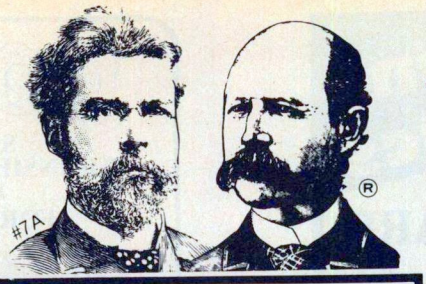
$39.50: Includes Peeks/Pokes Chart & Tip Book #4.

DRAW IN HI-RES on both Apple "pages" using easy keyboard commands OR paddles/joystick. Pre-view lines before plotting. Solid or mixed colors & Reverse (background-opposite) drawing. FAST one-keystroke circles, boxes & ellipses, filled or outlined. Add text for graphs & charts. All pix Save-able to disk, to be called from your Applesoft programs.

COMPRESS HI-RES DATA to 1/3 disk space (average) allowing more hi-res pictures per disk.

MANIPULATE IMAGES: Superimpose any two images, or RE-LOCATE any rectangular section of any drawing anywhere on either hi-res page.

HI-RES TYPE: Add text to your pictures with adjustable character-size and large-character color. Type *anywhere* with no Htab/Vtab limits. Type sideways too, for graphs. Includes Tip Book #4.



Beagle Bros
Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.*

(Don't Settle for Less!)

* DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who.

BEAGLE BASIC

APPLESOFT ENHANCER
by MARK SIMONSEN

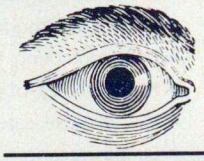
$34.95: Includes Peeks/Pokes Chart & Tip Book #6. Requires Apple IIe (OR II/II+ with RAM Card).

RENAME ANY APPLESOFT COMMAND or Error Message to anything you want. For program clarification, encryption/protection or even foreign translation. Plus add optional NEW COMMANDS:

ELSE follows If-Then statements, like this:
IF X=2 THEN PRINT "YES": *ELSE* PRINT "NO"

HSCRN reads color of any hi-res dot for collision testing. **SWAP X,Y** exchanges 2 variables' values. New **TONE** command writes music with no messy pokes & calls. **SCRL** scrolls text in *either* direction. **TXT2** lets Text Page 2 act exactly like Page 1.

PLUS: GOTO & GOSUB may precede variables, as in "GOSUB FIX" or "GOTO 4+X". Escape-mode indicated by special ESC CURSOR. Replace awkward Graphics screen-switch pokes with 1-word commands. Change ctrl-G Beep to any tone. **INVERSE REMS** too! All GPLE compatible.



```
1 FOR S=768 TO 773: READ A:
  POKE S,A: NEXT: POKE 232,0:
  POKE 233,3: DATA 1,0,4,0,5,0
2 HGR2: FOR R=0 TO 192: ROT=R:
  SCALE=96: XDRAW 1 AT 140,95:
  SCALE=30: XDRAW 1 AT 140,95:
  S=PEEK(49200): NEXT: RUN
```

PRONTO-DOS

HIGH-SPEED DOS / DOS-MOVE UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

TRIPLES THE SPEED of disk access and frees 10,000 bytes of extra memory by moving DOS.

Function	Normal	Pronto
BLOAD HI-RES IMAGE	10 sec.	3 sec.
LOAD 60-SECTOR PROGRAM	16 sec.	4 sec.
SAVE 60-SECTOR PROGRAM	24 sec.	9 sec.
BLOAD LANGUAGE CARD	13 sec.	4 sec.
(Text Files: No Change)		

Boot the Pronto disk or your updated disks, created with the normal INIT command. Compatible with all DOS Commands, GPLE, Double-Take, DOS Boss, DiskQuik and almost all unprotected programs.

MOVE DOS to your Language Card, RAM Card, or standard Apple IIe upper 16K, freeing up **10,000 EXTRA BYTES** of memory for your programs.

15 EXTRA SECTORS per disk. Catalog Free-Space displayed every time you catalog a disk.

TYPE-COMMAND ("TYPE filename") prints contents of sequential Text Files on screen or printer.

NEW!

DISKQUIK

DISK DRIVE EMULATOR
by HARRY BRUCE and GENE HITE

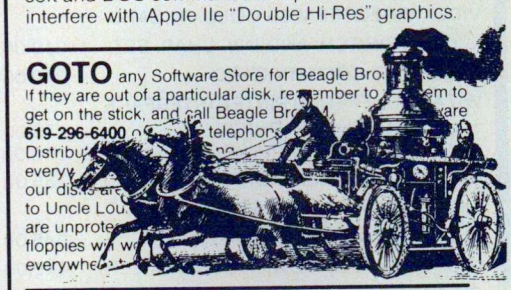
$29.50: Includes Peeks & Pokes Chart
Requires Apple IIe with Extended 80-column Card.

ACTS LIKE A DISK DRIVE in Slot 3, but much faster, quieter, more reliable and $350+ cheaper! Enjoy the benefits of a 2nd (or 3rd or 4th...) drive at less than 1/10th the price. Catalogs normally with "CATALOG, S3" command. Load & Save any kind of files into RAM with normal DOS commands.

SILENT AND FAST: Since no moving parts are involved, DiskQuik operates silently and at super-high speeds. See it to believe it. Your Apple IIe's Extended 80-column Card (required) can hold about half the amount of data as a 5¼" floppy disk!

MANY USES: For example, auto-load often-used files like FID etc., etc., into RAM when you boot up, so they are always available when you need them. Copy files from RAM onto disk and vice versa, just as if a disk drive were connected to slot #3.

FRIENDLY & COMPATIBLE with 80-column display, GPLE, ProntoDOS, and all normal Applesoft and DOS commands and procedures. Will not interfere with Apple IIe "Double Hi-Res" graphics.



GOTO any Software Store for Beagle Bro
If they are out of a particular disk, re member to em to
get on the stick, and call Beagle Bro M are
619-296-6400 o telephon
Distribu no
every
our dis
to Uncle Lou
are unprote
floppies wi
everywhe

DOUBLE-TAKE

2-WAY-SCROLL/MULTIPLE UTILITY
by MARK SIMONSEN

$34.95: Includes Peeks/Pokes AND Tips/Tricks Charts.

2-WAY SCROLLING: Listings & Catalogs scroll Up AND Down, making file names and program lines much easier to access. Change the Catalog or List scroll-direction at will, with Apple's Arrow keys.

80-COLUMN COMPATIBLE: All features support IIe and most other 80-column cards.

BETTER LIST FORMAT: Each program statement lists on a new line for *FAST* program tracing & de-bugging. Printer-compatible; any column-width.

VARIABLE-DISPLAY: Displays all of a program's strings and variables with current values.

CROSS-REFERENCE: Sorts and displays line numbers where each variable & string appears.

AUTO-LINE-NUMBER, Hex/Dec Converter, better Renumber/Append, Program Stats, Change Cursor, Space-On-Disk. GPLE/Pronto compatible.

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

Visa/MasterCard or COD, **call TOLL-FREE**
Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S Check, Money-Order or Visa MC# to **BEAGLE BROS**, 7th Floor
4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping, Any-Size Order.
Overseas add $4.00. COD add $3.00. California add 6%.
ALL ORDERS SHIPPED IMMEDIATELY

CLIP COUPON OR USE SEPARATE SHEET

Laing Electronics

10382 STANFORD AVE.
SUITE E-203
GARDEN GROVE, CA 92640
PHONE: (714) 534-4216

MICROSOFT 16K CARD
$59.00

★ HARDWARE FOR THE APPLE II ★

	LIST	YOUR COST
HAYES MICROMODEM II	379.00	258.00
HAYES SMARTMODEM 300 BAUD	289.00	199.00
HAYES SMARTMODEM 1200 BAUD	699.00	498.00
NOVATION APPLE CAT II	389.00	268.00
MICROSOFT Z-80 SOFTCARD CP/M	345.00	219.00
MICROSOFT PREMIUM PACKAGE (Z-80, VIDEX 80 COL., 16K CARD)	695.00	490.00
PCPI 6 Mhz APPLI-CARD (Z-80 CP/M)	375.00	279.00
VIDEX VIDEOTERM 80 COLUMN CARD (SOFTSWITCH, AND INV. CHAR. SET)	319.00	225.00
WIZARD 80 COLUMN CARD	249.00	149.00
WIZARD BUFFERED PINTER INTERFACE (16K BUFF + GRAPHICS)	179.00	129.00
(32K BUFF + GRAPHICS)	219.00	153.00
NEC HI-RES GREEN MONITOR (JB1201)	285.00	153.00
MICRO-SCI A2 APPLE COMPATIABLE DISK DRIVE W/O CONTROLLER	345.00	229.00
MICRO-SCI A2 DRIVE WITH CONTROLLER	445.00	299.00
OKIDATA MICROLINE 92/P PRINTER	699.00	498.00
PROWRITER 8510AP PRINTER	495.00	349.00
KENSINGTON SYSTEM SAVER FAN	89.95	65.00
GIBSON LIGHT PEN/LPS II (PENTRACK)	349.00	239.00
VISTA V1200 CART. DISK SYSTEM 6.2MB	1840.00	999.00
VISTA A-800 8" DISK CONTROLLER	379.00	249.00
VISTA SOLO 5¼" DISK DRIVE	299.00	195.00
8" DRIVE SUBSYSTEM WITH CONTROLLER 2.4MB	1900.00	995.00

★ VERBATIM DISKS ★

100 DATALIFE 5¼" (MD525-01-18158)	490.00	239.00
50 DATALIFE 5¼" (MD525-01-18158)	245.00	124.00

★ SOFTWARE FOR THE APPLE II ★

PFS: FILE (SOFTWARE PUB.)	125.00	76.00
PFS: REPORT (SOFTWARE PUB.)	125.00	76.00
PFS: GRAPH (SOFTWARE PUB.)	125.00	76.00
DB MASTER REG. VERSION (STONEWARE)	229.00	152.00
DB MASTER VERSION #4 (STONEWARE)	350.00	231.00
MULTIPLAN (MICROSOFT) CP/M OR DOS3.3	275.00	167.00
WORDSTAR (MICRO-PRO CP/M)	395.00	327.00
WORDSTAR WITH Z-CARD (PCPI APPLI-CARD)	495.00	425.00
SCREEN WRITER II (ON-LINE)	129.95	86.00
FORTRAN-80 (MICROSOFT CP/M)	195.00	129.00
ASCII EXPRESS PROFESSIONAL	129.95	85.00
Z-TERM PRO. (SOUTHWESTERN DATA)	149.95	98.00

If Personal Check, Allow
10 BANKING DAYS for CLEARANCE.
Prices and specifications subject to change without notice. Unless otherwise noted,
ALL PRICES are PLUS SHIPPING.
SHIPPING & HANDLING—ADD $5.00 UPS SURFACE (DISK) DRIVES $7.50—MONITORS $10.00—PRINTERS $15.00.
California residents add 6% Sales Tax.

Circle 9 on Reader Service Card

COOL-AIDE™

SUPER COOLING FAN
TRANSIENT VOLTAGE PROTECTION
RFI/EMI FILTERING
FOR APPLE II* COMPUTERS
$59.95

Features

- Thin, compact design, easy installation, just clips on
- Compatible with Apple standard computer case
- Entire system controlled by front power switch, 120/60 operation
- Internal voltage surge protection & RFI/EMI filtering
- Power indicator light & dual auxiliary outlets
- Quiet and efficient operation
- Reduce heat build up, moves 37 CFM of air

Also available from Kalglo Electronics Co. Inc.:
AEGIS™ Power Conditioning Equipment
SPIKE-SPIKER® - Transient voltage protectors & noisefilter *from* $34.95-$94.95
LINE-SAVER™ - Uninterruptable Power Systems - *from* $395.00

SEND FOR FREE LITERATURE

MasterCard VISA AMERICAN EXPRESS

KALGLO®

6584 Ruch Rd., Dept. CA
Bethlehem, PA 18017

Use your credit card or send check and we pay shipping
Out of state, order toll free
800-523-9685 TWX 510-651-2101
215-837-0700
Pa. Res. add 6% sales tax
for COD add $3.00 + shipping
DEALER-OEM INQUIRIES INVITED
*Reg. trademark of Apple Computers Inc.

Circle 23 on Reader Service Card

TIRED OF PAYING BILLS?

Then let your Apple do it! The newly-enhanced and improved BILL PAYER™ System turns your Apple into a complete check-writing and record-keeping system for all those bills that you have to pay on a regular basis. Its powerful features include:

- A professional-looking check (check order forms included).
- All files & program on one disk.
- Fully prompting — easy to use.
- Maintains list of payees — pays bills QUICKLY!
- 99 definable categories for each payment.
- Built-in reconciler — almost balances itself!
- Search & total by payee and category.
- Checks formatted for window evelopes.
- Uses Diversi-DOS™ for super-fast disk access.
- Disk is copyable for easy backup.

System requirements: Apple II+ or //e, 48K, disk drive, DOS 3.3, printer.

Send check or money order for **$49.95** to:

PLEASURE POINT ENTERPRISES
5483 PLEASURE POINT LANE
BELLEVUE, WA 98006

Washington State residents add $3.95 (7.9%) sales tax.

Apple, Apple II+, Apple //e, DOS 3.3 are trademarks of Apple Computer, Inc. Diversi-DOS is a trademark of DSR, Inc.

Circle 33 on Reader Service Card

TELECOMMUNICATIONS CONFERENCE

Michael Thyng

ON Saturday evening, October 15th, 1983 I attended the Telecommunications User's Group meeting at the University of Washington. About 80 members and guests showed up to hear Mr. Jerry McCallom, Marketing Representative for Pacific Northwest Bell Telephone Company, explain upcoming changes. I think most of us were there to get answers to the rumors we had been hearing on possible new and separate charges for modems.

For perspective, McCallom offered a bit of historical background, then explained what we can expect in the way of changes, come January, 1984. Originally, one large company, American Telephone and Telegraph (AT&T), controlled both local and long distance calls and had a great deal of influence is setting local and long distance rates. Commencing in January 1984, AT&T will be one of many long distance phone companies. There will be 22 distinct, separate, independent local phone companies. In my area, that company is Pacific Northwest Bell.

Where one huge company supervises an entire system, dollars from the more profitable areas of operation can be applied to offset losses and support the not-so-profitable sections. Well, the profitable section is long distance. The not so profitable section is local calling. Think for yourself. If you take away the profit center and leave the not-so-profitable, what do you have to do if you're going to stay in business? All those who said raise the rates got it right.

We have been hearing about rate changes almost continuously since the government ruled AT&T must be broken up. McCallom cautioned us to realize that with 22 separate local companies, you could have 22 different ways of charging for phone service depending on where you lived. As we hear in the news about this or that local company's charges, don't confuse it as additional charges by your local company. They may or may not be.

So now we get down to brass tacks. Do we or don't we get charged for modems? Some local phone companies are planning on charging for modems. Some plan a flat rate. Some plan a connect charge. Only your local company can tell you for sure.

Pacific Northwest Bell is planning modem charges for businesses. I asked McCallom what the charges might be or how they would be assessed, but he did not have that information. Some attendees lined up to ask about the possibility of modem charges for the home user. He told us that Pacific Northwest Bell had no plans at this time to charge private modem users.

There was a contingent of hearing impaired modem users who were concerned that modem charges may be imposed without regard to their needs. Where the non-handicapped may use the modem for recreational use, the hearing impaired may only be able to get some information through their modem. Again, with no plans for modem charges, McCallom couldn't say whether there would be reduced or suspended fees for any groups if modem charges are ever imposed on home users.

The question comes up "why should modems be charged any differently from regular phone calls?" As I understand it, it works like this. Your voice call is very slow. When you and I are talking, our voices are routed and rerouted around various service lines — whichever line is available at a particular instant. When we are talking, we will never notice it. We talk. We pause. But a modem requires constant "on" to maintain a connect condition. That constant connect cannot easily be rerouted. More precisely, that modem using line cannot be used for some other conversations during the pauses because there are no pauses. So the modem ties up a line. More use than regular voice. More modems, more use. More use for modems, fewer lines for voice. More need for equipment. More money needed...oops, here we go again, back into a full-duplex loop.

BYTES & PIECES PRICE RIOT

Ask For Free Price List

$1.00 credit for phone orders over $100.00

	LIST	SPECIAL
Alaska Card	129.95	107.19
Bank Street Writer	69.95	49.96
Bookends	124.95	90.46
BurgerTime	34.95	23.46
D.B. Master 4.0	350.00	252.96
Ess. Data Dup. (EDD)	79.95	58.16
Einstein Compiler	129.00	93.46
Exodus (Ultima III)	54.95	39.66
Home Accountant	74.95	53.46
Incredible Jack	179.00	130.46
Koala Pad	124.95	90.46
Krell Logo	89.95	75.46
Krell SAT	299.95	252.96
Micro Cookbook (II+ or IIe)	40.00	28.56
New Step by Step	89.95	64.56
Nibbles Away II	69.95	49.76
Planetfall	49.95	35.46
Replay II (II+ or IIe)	130.00	107.19
Sensible Speller	125.00	90.46
16K Card (Microsoft)	99.95	75.96
S'Bear Opposites	39.95	28.56
Super Text Pro	99.00	72.96
Think Tank	150.00	108.96
Verbatim Disks (Library Case)	49.00	29.50
Verbatim Disk Drive Analyzer	69.00	56.69
Word Handler	59.95	42.86

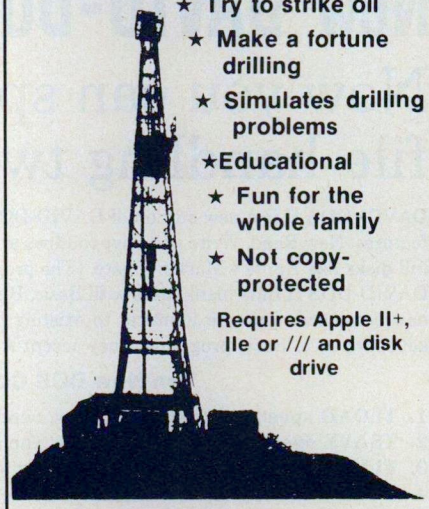
Ask for our educational software catalog

C.O.D. • M.O. • Cert. Checks • School P.O. • VISA • MasterCard • Checks Allow 2 Weeks • N.Y.S. Res. Add Sales Tax • US Orders Under $150. Add $2.00 P & H • All Canadian, US Funds $3.00 P & H. Foreign, Charges Only, Min. P & H $6.00

Box 525, Dept. C, East Setauket, N.Y. 11733
(516) 751-2535

WILDCATTER

★ Computer game with HIRES graphics



★ Try to strike oil
★ Make a fortune drilling
★ Simulates drilling problems
★ Educational
★ Fun for the whole family
★ Not copy-protected

Requires Apple II+, IIe or /// and disk drive

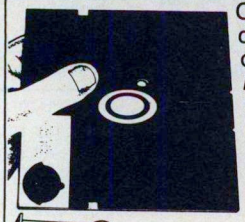
$50.00
includes postage and handling.
+A.P.P.L.E. Members **$31.00**

Merlin Software
4300 Houma Blvd., Suite 108
Metairie, LA 70002
(504) 885-9941
Visa - Mastercard

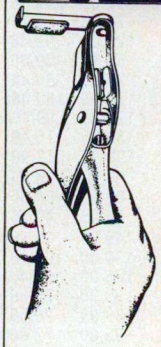
Apple II+, IIe & /// trademarks of Apple Computer.

Circle 34 on Reader Service Card

*REDUCES DISKETTE COST 50%!
DOUBLES DISKETTE STORAGE SPACE



Owners of 5¼" *single *"read/write"* head disk drives can immediately double diskette storage space by using ***NIBBLE NOTCH I*** or ***II.*** The back of single sided diskettes are burnished. To use it you need a *"write enable notch,"* and some also need in addition, an *"index hole."* ***NIBBLE NOTCH I*** and ***II*** are precision engineered tools designed exclusively for this purpose.



NIBBLE NOTCH I

(Cuts square Write Enable Notch) For users of Apple, Franklin, Kaypro, Atari, Commodore and most other soft sectored systems.

only **$14.95** each

add $1.50 each P&H / ($4.50 foreign P&H)

NIBBLE NOTCH II

(Cuts square Write Enable Notch and ¼ inch round *"index hole"*) for owners of IBM, TRS 80 I & III, Osborne and others needing *"index hole"*, and all other *"hard sectored"* systems.

only **$21.90** each

add $2.50 each P&H / ($6.50 each foreign P&H)

Florida Residents Add 5% Sales Tax

• IT'S A MONEY SAVER! •

Sold at leading computer and office supply stores in your area.

Dealer inquiries are invited.

ORDER TODAY!

SEND CHECK OR MONEY ORDER TO:

NIBBLE NOTCH™

DIVISION OF CORTRAN INTERNATIONAL

4211 N.W. 75 TERRACE • DEPT. 76 • LAUDERHILL, FL 33319

PAT. PEND. ALL TRADEMARKS ARE ACKNOWLEDGED

Circle 20 on Reader Service Card

GRAPHMAX™

GRAPHICS PRINTER CARD FOR YOUR APPLE

COMPATIBLE WITH GRAPPLER+ BUT FAR MORE POWERFUL!

EXTRAS YOU GET

- **ZOOM MAGNIFICATION** up to **99** times with
- full **PICTURE CROPPING**
- user **UPGRADABLE** for **COLOR** printers Or available with color option.
- up to 200 characters of **HEADER**/page
- up to 200 characters of **FOOTER**/page
- auto **PAGE NUMBERING**
- versatile **IMAGE LOCATION**
- improved **ASPECT RATIOS** for graphics
- bigger selection of printers
- better engineered & backed by full 2-year warranty

PLUS HAS ALL THE STANDARD FEATURES OF GRAPPLER+

micromax™

INNOVATORS IN MICRO-COMPUTER TECHNOLOGY

MICROMAX SYSTEMS, INC. 6868 NANCY RIDGE DRIVE, SAN DIEGO, CA 92121

619-457-3131

159.95

Suggested Retail

DOLLAR FOR DOLLAR, IT'S A

MUCH BETTER BUY

GRAPPLER+ is a trademark of Orange Micro, Inc.

Circle 27 on Reader Service Card

New DAVID-DOS II™ Compared With Other DOS's

Now you can speedup Text-file handling two ways

DAVID-DOS II is a new edition of DAVID-DOS with added speed, commands, and features. New Read, Write and Save routines are high speed. DAVID-DOS II updates full disks like Apple's Master Create. (The programs on your disks are not touched). DAVID-DOS II Inits blank disks with Basic, Binary or Exec HELLO in seconds. Ten new commands operate identical to existing DOS commands. Use them from the keyboard or in Basic programs. They accept A & L parameters.

Ten New DOS Commands

1. **TLOAD speed loads all Text Files, random or sequential, to ram.**
2. **TSAVE speed saves all Text Files, random or sequential, from ram.**
3. **TLIST Lists all Text Files, random or sequential to screen/printer.**
4. **DUMP Memory to screen/printer in Hex with Ascii on right side.**
5. **DISA disassembles Binary to screen or printer.**
6. **AL prints last loaded program Address & Length in decimal & hex.**
7. **HIDOS moves DOS to Language Card & continues operation of program.**
8. **/ is a one keystroke Catalog in addition to the original command.**
9. **DATE prints with any clock. Also File Dating with clock or manual.**
10. **FIND prints address's of hex found in 64k memory. Hidos cmd only.**

Compatible

All DOS entry addresses have been preserved. DOS is original length and compatible with most software. David-Dos II is copyable and creates fully copyable updated disks. DAVID-DOS II is licensed by programmers for inclusion in the software they sell. Init areas were used for David-Dos II. Works with all Apple IIs including IIe 80 Col, Franklin & Hard Disks, such as Corvus & Xebec. Requires 48K. Complete documentation for screen or printing and many utilities are on the disk.

All times in seconds. (Time Test programs available)		DAVID DOS-II	ProDOS	DIVERSI DOS	DOS 3.3
TEXTFILES (100 Sectors)	TSAVE	8.0	NO	NO	NO
	TLOAD	6.2	NO	NO	NO
(791 Strings,	WRITE	29.3	28.0	29.4	88.4
32 chars ea)	READ	24.3	16.3	24.3	83.8
	PRINT/READ	44.2	45.9	45.1	117.1
(442 Sectors, 7 x 500)	APPEND	142.3	142.9	151.1	1231.2
APPLESOFT (100 Sectors)	*SAVE	7.1	16.4	6.4	33.1
	LOAD	5.0	4.0	5.0	23.5
INTEGER (100 Sectors)	*SAVE	7.3	NO	6.6	33.4
	LOAD	4.9	NO	4.9	23.4
BINARY (100 Sectors)	*BSAVE	7.8	18.4	7.3	28.7
	BLOAD	5.8	4.8	5.8	24.5
48K PROGRAM SPACE	APPLESOFT	36,352	NO	36,352	36,352
(With 3 Bufs avail)	INTEGER	36,352	NO	36,352	36,352
	BINARY	36,352	34,816	36,352	36,352
64K PROGRAM SPACE	APPLESOFT	46,592	31,232	45,658	35,162
(With 5 Bufs avail)	INTEGER	46,592	NO	35,162	35,162
	BINARY	46,592	40,704	45,658	35,162
NUMBER OF DOS COMMANDS		37	29	31	28
CLOCK FILE DATING		YES	YES	NO	NO
MANUAL FILE DATING		YES	NO	NO	NO
ONE KEYSTROKE CATALOG		YES	NO	YES	NO
AUTO USE INTEGER CARD ANY SLOT		YES	NO	NO	NO

Each Program was tested twice W/Apple Clock Card on a newly formatted disk containing DOS.
*Add 5 seconds for Verify. Apple II, Applesoft & ProDOS are trademarks of Apple Computer

To Order: Send Check or Phone Visa/MasterCard.
Add $2.00 Shipping. Overseas add US $4.00
Calif. add 6%. All orders airmailed.

$39.95

DAVID DATA™

12021 WILSHIRE BLVD., SUITE 212A
LOS ANGELES, CA 90025 (213) 478-7865

Circle 7 on Reader Service Card

1983 CALL —A.P.P.L.E. INDEX

Compiled by Kathryn Hallgrimson Suther/Val J. Golding

TITLE	AUTHOR	ISSUE	PAGE
ADVANCED TECHNICAL			
Applesoft Can Throw Your Valuables in the Trash	Cornelis Bongers	May	35
Arcade Graphics	Chris Jochumson	Apr	9
Creating an Amper-Library	Bill Parker	Jun	23
Data Encryption Techniques	James L. Paul	Apr	47
Data Process, The	Ken Kashmarek	Dec	13
DOS Continues Following This Interruption	O.J. Sizemore	Dec	43
DOS Self-Synch Demo, A	John Webster	Oct	77
Down the Data Path	Ken Kashmarek	Nov	17
80 Column //e Low-Res Graphics	Rob Moore	Jul	9
Fast 6502 Multiplication	Damon Slye	Jun	45
Fixes for the Apple //e DOS 3.3	Cecil Fretwell	Dec	41
Interrupts and Apples	Richard L. Emerson	Feb	35
Joys of Relocatability, The	Bill Parker	Apr	27
New Pseudo Random Number Generator	John Russ	Jan	33
Painting the RAMcard	Donald W. Miller, Jr.	Apr	51
RND is Fatally Flawed	David Sparks	Jan	29
Storage Access	Ken Kashmarek	Oct	17
Switchcraft: Memory Management	Coleman Logan	Mar	9
Taking a Bigger Byte	Ken Kashmarek	Feb	27
Wait-ing in the Monitor	John K. Elberfeld	Mar	75
What's the DIF?	David G. Sparks	Nov	29
Where has the Data Gone?	Ken Kashmarek	Sep	11
Winner is, And The	Ken Kashmarek	Sep	73
APPLICATIONS			
Alternative Text Page, An	Glen Bredon	Mar	67
ApMail Apple-cations	Ken Kashmarek	Sep	78
Apple in the Kitchen, An	Jim Ganz	Feb	43
APPLE — The Magician	Luis Hernandez	Mar	53
Complete Password System, A	Jason K. Mintz	May	27
Creating an Amper-Library	Bill Parker	Jun	23
Data Encryption Techniques	James L. Paul	Apr	47
80-Col. //e Low-Res Graphics	Rob Moore	Jul	9
Exploiting TASC	Sam F. Vass	Dec	55
Hi-Res Slide Show	Stowe Keller	Dec	49
Magazine Database, A	Clay Carr	Nov	79
Magfiler	Jack Olivieri	May	19
Memory Moves Made Simple	Ken Kashmarek	Jun	53
Mockingboard Speech Editor	Clay Ruth	Oct	7
Name Matchcode Search	Lorin Hungsberg	Aug	49
Password Protection	Ralph H. Swerdlow	Jan	13
Soft 70 with D.C. Hayes	Gene Hite	Feb	63
Spelling Corrector	Wesley R. Felty	May	51
Telecommunications with Filedump	Steve Ross/Val Golding	Sep	84
Textfile Scrambler	Skene H. Moody	Apr	49
Trig, Degrees/Common Logs	W. S. Skeen	Apr	60
BUSINESS			
Data Analysis	Jeffrey Jacques Ph.D	Dec	21
Data Process, The	Ken Kashmarek	Dec	13
Down the Data Path	Ken Kashmarek	Nov	17
Extra Memory for THE Spreadsheet	George Mochizuki	Sep	37
Spreadsheets in the Classroom	Jeffrey Jacques, Ph.D	Oct	43
Spreadsheets in the Classroom	Jeffrey Jacques, Ph.D	Nov	41
Spreadsheet Programs	Steve Ross	Jun	17
Spreadsheet Video Terminal Driver	Glenn Ferreri	Oct	49
Strictly Business	Mike Regeimbal	Nov	37
Strictly Business	Mike Regeimbal	Dec	27
Storage Access	Ken Kashmarek	Oct	17
What's the DIF?	David Sparks	Nov	29
Where has the Data Gone	Ken Kashmarek	Sep	11
CPM			
Andromeda RAM Card	Jay Lieske	Aug	61
Fix APCPM	Murray Arnow	Mar	111
Nibbling Away at CP/M	Murrary Arnow	Jun	49
Printer Interfacing	Murrary Arnow	Nov	61
Softcard LST Driver, A	Gerald Cahill	Sep	69
Why CP/M	Murrary Arnow	Oct	59
DOS			
DOS Continues Following This Interruption	O.J. Sizemore	Dec	43
DOS Self-Synch Demo, A	John Webster	Oct	77
Fixes For the Apple //e DOS 3.3	Cecil Fretwell	Dec	41
Taking the "Pro Out of ProDOS	G. Charpentier/ D. Sparks	Nov	9

TITLE	AUTHOR	ISSUE	PAGE
EDUCATION			
Apple's Core, The	Candy Woods	Aug	53
Apple for Nellie, An	Rick Sutcliffe	Oct	29
Apple for the Teacher, An	Kathryn H. Suther	Aug	48
Apple for the Teacher, An	Kathryn H. Suther	Sep	39
Apple for the Teacher, An	Kathryn H. Suther	Oct	27
Apple for the Teacher, An	Kathryn Suther	Dec	37
A.P.P.L.E. Computer Camp '83	Charlie/Susan Stillman	Sep	43
Apple /// Revolutionizes Braille Translation	Evelyn Iritani	Dec	35
Arcademic Skill Builders	Kathryn H. Suther	Jul	35
Arcadian Approach, The	Michael Shadick	Jun	31
BASIC vs Logo	Lee Reynolds	Sep	15
Big Adding Machine, The	Gary Leonard	Oct	35
Big Reducer, The	Gary Leonard	Nov	45
Computers in Education '83	K. Suther/C. Stillman	Aug	51
Education is an Adventure	Dr. Mark Cross	Jun	35
Flying High	Earl Keyser	Oct	55
From LISP to Logo	Al Gruber	Aug	13
Grade Book Review	W. Felty/P. Buck	Sep	39
Kids and the Apple	Kathryn H. Suther	Jun	37
Magic Square Dance	J. Taylor	Sep	51
Master Type	Candy Woods	Jun	39
Numbers	E. Stephen Freeman	Aug	55
Pre School Alphabet	Garry Margolis	Nov	48
Rocky's Boots	Earle Hancock	Sep	47
Spanish for the Traveler	Helen Hamerstrom	Oct	39
Spreadsheets in the Classroom	Jeffrey Jacques, Ph.D	Oct	43
Spreadsheets in the Classroom	Jeffrey Jacques, Ph.D	Nov	41
30 Crates of Apples	Wes Felty	Jul	31
Tutor, Tool & Tutee	Eric Flescher	Dec	34
GRAPHICS			
Applesoft Graphics Mover	Homer O. Porter	Sep	
Arcade Graphics	Chris Jochumson	Apr	9
Ceemac	Jim Hockenhull	Sep	21
80-Col //e Lo-Res Graphics	Rob Moore	Jul	9
Graphics Magician, The	Harvey Gratt	Sep	25
Hi-Res Slide Show	Stowe Keller	Dec	49
Printographer, The	Roger Lilly	Nov	53
Quick Zapper	S. Keller/V. Golding	Oct	52
Shape Table Splicer	Cyrus W. Roton	Sep	33
Smooth Animation	Jonathan Kandell	Feb	61
Special Effects	Jeff Harper	May	43
Splitter	Norman L. Kushnick	Jan	53
Three Dee Demos	David Sun	Jan	49
HUMOR			
Boot	Craig MacIntosh	Jan	5
Boot	Craig MacIntosh	Feb	7
Boot	Craig MacIntosh	Mar	5
Boot	Craig MacIntosh	Apr	5
Boot	Craig MacIntosh	May	7
Boot	Craig MacIntosh	Jun	7
Boot	Craig MacIntosh	Jul	5
Boot	Craig MacIntosh	Aug	5
Boot	Craig MacIntosh	Sep	5
Boot	Craig MacIntosh	Oct	7
Boot	Craig MacIntosh	Dec	5
So What Did You Expect?		Apr	28,63,64
So What Did You Expect?	Ken Kashmarek	Jun	72
So What Did You Expect?	Garth Baulch	Jul	7
So What Did You Expect?		Sep	10,52, 76,95,96
So What Did You Expect?		Oct	56,75
So What Did You Expect?	P.F. Steele/E.P. Darragh	Nov	47,68
So What Did You Expect?		Dec	20,26, 42,78
LANGUAGES			
BASIC vs Logo	Lee Reynolds	Sep	15
Catching Mumps	David Walker	Apr	23
Ceemac	Jim Hockenhull	Dec	9
"C" What I Mean?	Jim Hockenhull	Dec	9
Flying High	Earl Keyser	Oct	55
From LISP to Logo	Al Gruber	Aug	13

1983 CALL —A.P.P.L.E. INDEX

TITLE	AUTHOR	ISSUE	PAGE
MISCELLANEOUS			
Age of Silicon	Jonathan Post	Sep	7
American Flag	Alan C. Young	Jul	60
Another Use for the Apple	Bruce Bourbonnais	Sep	87
Apple Art	Jack Bender	Jul	38
Apple Blossoms	Val J. Golding	Jul	39
Apple Developer's Conference	Dave Redhead	Aug	59
A.P.P.L.E. Doctor	George Spelvin	Feb	65
A.P.P.L.E. Doctor	George Spelvin	Apr	65
A.P.P.L.E. Doctor	David Sparks	May	75
A.P.P.L.E. Doctor	Bruce Field	Jul	55
A.P.P.L.E. Doctor	Bruce Field	Aug	67
A.P.P.L.E. Doctor	Bruce Field	Sep	65
A.P.P.L.E. Doctor	Bruce Field	Oct	63
A.P.P.L.E. Doctor	Bruce Field	Dec	65
APPLE - The Magician	Luis Hernandez	Mar	53
Automated Typesetting	Michael Shadick	May	37
Birds and the Bees, The	Rick Sutcliffe	Oct	61
Birds and the Bees, The	Rick Sutcliffe	Nov	63
Birds and the Bees, The	Rick Sutcliffe	Dec	59
Bored with Bulletins	Michael Thyng	May	45
Call -A.P.P.L.E. Quickie, A		Aug	61
Call -A.P.P.L.E. Quickie, A	Ken Kashmarek	Oct	17
Comparing Apples and Apples	Cecil Fretwell	Jun	9
Computer Home, The	Burt Webb	Apr	17
Confessions of a Cheapskate	A. Floeter/V. Floeter	Jul	43
Conjurer's Square	J. Taylor	Nov	59
Data Encryption Techniques	James L. Paul	Apr	47
Disease Was Her Key to Success	Michael Shadick	Oct	66
Error Message Dictionary	Lee Webster	Mar	59
Exotic Technologies	Jonathan Post	Nov	23
From LISP to Logo	Al Gruber	Aug	13
Gaming Room, The	Val J. Golding	Sep	80
Gaming Room, The	Val J. Golding	Oct	72
Getting the Point	Michael Thyng	Jan	63
How To Become An Author	Daniel T. Lyons	Sep	82
Keep it Clean	Jim Hockenhull	Oct	53
Live Wire, A	Val J. Golding	Jan	7
Magazine Database, A	Clay Carr	Nov	79
Magic Square Dance	J. Taylor	Sep	51
Making it Last	Robert Hanson	Mar	99
Man, Machine and the Anagram	Donald W. Miller, M.D.	Apr	31
Microsolitude	Michael Shadick	Mar	78
Mockingboard Reset Modification	Michael Mahon	Dec	45
Monster Lives, The	Michael Thyng	Oct	65
Nellie and the Pirates	Rick Sutcliffe	Sep	55
Nellie's Formatter	Rick Sutcliffe	Aug	63
1982 Call -A.P.P.L.E. Index		Jan	66
1984 is coming. . . But Not Coming True!	Michael Shadick	Aug	57
Northern Spy	Rick Sutcliffe	Jun	57
Northern Spy	Rick Sutcliffe	Jul	47
Out-Wizarding Wizardry	Christopher H. Williams	Dec	61
Pacific Science Center Festival	Michael Thyng	May	74
Parkinson's Law	Robert Greenwald	Jul	53
Removing the Challenge	Luis Hernandez	Apr	35
Robotics	Michael Thyng	Mar	115
Robotics to Graphics	Jonathan Post	Oct	23
Saga of Greg Grummore, The	David Sparks	Apr	39
Tomorrow's Apples Today		Jan	77
Tomorrow's Apples Today		Feb	71
Tomorrow's Apples Today		Mar	118
Tomorrow's Apples Today		Apr	66
Tomorrow's Apples Today		May	67
Tomorrow's Apples Today		Jul	72
Tomorrow's Apples Today		Aug	71
Tomorrow's Apples Today		Sep	91
Tomorrow's Apples Today		Oct	69
Tomorrow's Apples Today		Nov	71
Tomorrow's Apples Today		Dec	71
TRS Connection, The	Art Ude	Nov	55
Two Views of the Micro God?	Michael Shadick	Dec	36
Visit with Cornelis Bongers, A	Ralph Swerdlow	Mar	47
What Do I Do Now Daddy?	David Sparks	Oct	79
Where Have All The Typists Gone	Gary Breidenstein	Jun	41
Wizardry and How to Beat it	Art Ude	Aug	43
Word, The	Michael Shadick	Mar	113
Write A.P.P.L.E.	The Readers	Jan	69
Write A.P.P.L.E.	The Readers	Feb	66
Write A.P.P.L.E.	The Readers	Apr	57
Write A.P.P.L.E.	The Readers	May	61
Write A.P.P.L.E.	The Readers	Jun	73
Write -A.P.P.L.E.	The Readers	Jul	67
Write A.P.P.L.E.	The Readers	Aug	75
Write A.P.P.L.E.	The Readers	Sep	89
Write A.P.P.L.E.	The Readers	Oct	67
Write A.P.P.L.E.	The Readers	Nov	66
Write A.P.P.L.E.	The Readers	Dec	68
Writing the Wrongs	the wrongdoers	Apr	72
Writing the Wrongs	the wrongdoers	May	71
Writing the Wrongs	the wrongdoers	Jun	59,66
Writing the Wrongs	the wrongdoers	Jul	97,98
Writing the Wrongs	the wrongdoers	Aug	78
Writing the Wrongs	the wrongdoers	Oct	74
Writing the Wrongs	the wrongdoers	Nov	65
You Think You Have A Million Dollar Program	Alan D./ Valerie A. Floeter	May	9
PASCAL			
Beginners Guide To the Pascal Editor, A	Terry Reilly	Mar	89
Getting A Quart into a Pint Pot	Chris W. Green	May	13
Inside Story, The	Chris W. Green	Jul	21
Pascal Magic-A Better Keypress	J. Buckley/R. DeGroat	Jun	65
Pascal 1.2	Allen Munro	Sep	77
Seeing Double Safely	Lawrence O'Glasser	Mar	79
System.Library	David Walker	Sep	75
Things Have Gotten Better With Time	Bruce van Norman	Aug	47
What in the Blaise is Going On?	Bruce van Norman	Jun	63
REVIEWS			
Ampersand Utilities	David Sparks	Mar	37
Apple's Core, The	Candy Woods	Aug	53
Apple Flasher	Walter Hopkins	Mar	71
Apple For the Teacher	Kathryn H. Suther	Dec	37
Arcademic Skill Builders	Kathryn H. Suther	Jul	35
Catching Mumps	David Walker	Apr	23
Ceemac	Jim Hockenhull	Sep	21
ChequeMate Plus	Michael Thyng	Dec	62
Comparing Apples and Apples	Cecil Fretwell	Jun	9
"C" What I Mean?	Jim Hockenhull	Dec	9
Computer Work Station	Val J. Golding	Oct	75
Dietician, The	Joan Chaiten	Feb	57
Enhancing Your Apple	Jim Hockenhull	Jul	59
Game of Life, The	Val J. Golding	Jul	65
Gaming Room, The	Val J. Golding	Oct	72
General Manager, The	Doug Pearson	Jun	43
Grade Book Review	W. Felty/P. Buck	Sep	39
Graphics Magician, The	Harvey Grattt	Sep	25
Health Aide	Richard Wahl	Nov	51
"I Sing the Apple Electric"	William C. Dwyer	Jun	61
Kids and the Apple	Kathryn H. Suther	Jun	37
Master Type	Candy Woods	Jun	39
MX Plus	Drew Berentes	Jan	41
Printographer, The	Roger Lilly	Nov	53
Reving Up the Rev e	C. Howard, M.D./ A. Schumer	Feb	9
Rocky's Boots	Earle Hancock	Sep	47
Spanish for the Traveler	Helen Hamerstrom	Oct	39
Special Effects	Jeff Harper	May	43
Spreadsheet Programs	Steve Ross	Jun	17
Taking the "Pro" Out of ProDOS	G. Charpentier/ D. Sparks	Nov	9
Time Manager, The	John Holmes	Aug	39
Transforth	Charles Wells	Mar	13
TRS Connection, The	Art Ude	Nov	55
Tutor, Tool & Tutee	Eric Flescher	Dec	34
VB-3 Microkeyer	Jim Hockenhull	May	57
Window	Michael Thyng	May	39
TECHNIQUE			
An Alternative Text Page	Glen Bredon	Mar	67
Apmail Applications	Ken Kashmarek	Sep	78
Applesoft Can Throw Your Valuables in the Trash Can	Cornelis Bongers	May	35
Arcade Graphics	Chris Jochumson	Apr	9
Beep: A Tale of (T) Error	Christopher Volpe	Mar	114
"Betcha Dollars to Apples.."	Peter C. Lincoln, Ph.D	Jul	15
Bottoms Up!	Tom Little	Jan	17
Branch Speeding	R.L. Sakrison	Feb	53
Controlling Control D	Ralph Swerdlow	Feb	59
Converting "NEW" to "FP" Type Files	W. T. Doyle	Mar	105
Creating an Amper-Library	Bill Parker	Jun	23
Customizing GPLE Escape Tables	Edward Nutter	Mar	93
Data Encryption Techniques	James L. Paul	Apr	47
Delete That Incubus!	Howard S. Balsam	Mar	107
Do Nothing 'Till You Hear From Me		Apr	63
DOS Self-Synch Demo, A	John Webster	Oct	77
Double Monitors for the Apple][	Greg DeWilde	May	47
80 Column //e Low-Res Graphics	Rob Moore	Jul	9
Escape From the Motherboard	Donal W. Miller, M.D.	Mar	51
Exploiting TASC	Sam F. Vass	Dec	55
Fast 6502 Multiplication	Damon Slye	Jun	45
Fix APCPM	Murray Arnow	Mar	111
Grappler + MX-80 with Graftrax	Ken Kashmarek	Jul	62

TITLE	AUTHOR	ISSUE	PAGE
Hi-Res Slide Show	Stowe Keller	Dec	49
Ignoring 7Extra Ignored	Ari/Dov Freund	Sep	61
Inverse String Printer	Stowe Keller	Sep	81
Joys of Relocatability, The	Bill Parker	Apr	27
Literal Input	Greg Grummore	Apr	43
Memory Moves Made Simple	Ken Kashmarek	Jun	53
Mockingboard Reset Modification	Michael Mahon	Dec	45
Mockingboard SpeechEditor	Clay Ruth	Oct	7
Painting the RAMcard	Donald W. Miller,Jr.	Apr	51
Preparing a Menu	Val J. Golding	Sep	68
Quotable Quote, The	Val J. Golding	Apr	55
RAMfinder	Mitchel Pilot	Aug	17
Smooth Animation	Jonathan Kandell	Feb	61
Soft 70 with D.C. Hayes	Gene Hite	Feb	63
Splitter	Norman L. Kushnick	Jan	53
Spreadsheet Video Terminal Driver, The	Glenn R. Ferreri	Oct	49
Storage Access	Ken Kashmarek	Feb	27
Switchcraft: Memory Management	Coleman Logan	Mar	9
Taking a Bigger Byte	Ken Kashmarek	Feb	27
Taming the IIe	Sandy Mossberg	Aug	9
Technique Notes	Ken Kashmarek	Dec	46
Textfile Scrambler	Skene H. Moody	Apr	49
Three Dee Demos	David Sun	Jan	49
Time is Now	Peter D. Hipson	Aug	23
Trig, Degrees/Common Logs	W.S. Skeen	Apr	60
Two "Trick" Programs	Lee Reynolds	Apr	61
Waiting for a Keypress	Ken Kashmarek	Mar	77
Wait-ing in the Monitor	John K. Elberfeld	Mar	75
Where to put Machine Code	Bill Parker	Jan	39
Winner is, And the	Ken Kashmarek	Sep	73
Wizardry and How to Beat it	Art Ude	Aug	43
TUTORIALS			
Anatomy of an Eamon Adventure	Robert Plamondon	Mar	21
Arcade Graphics	Chris Jochumson	Apr	9
Beginner's Guide	Lee Webster	Jan	9
Beginner's Guide to the Pascal Editor, A	Terry Reilly	Mar	89
Controlling Control D	Ralph H. Swerdlow	Feb	59
Data Encryption Techniques	James L. Paul	Apr	47
Digital Clock	Ralph Swerdlow	Aug	32
Error Message Dictionary	Lee Webster	Mar	59
Memory Moves made Simple	Ken Kashmarek	Jun	53
Password Protection	Ralph H. Swerdlow	Jan	13
Preparing A Menu	Val J. Golding	Sep	68
Quotable Quote, The	Val J. Golding	Apr	55
Taking the "Pro" out of ProDOS	G. Charpentier/ D. Sparks	Nov	9
UTILITIES			
Catalog Clock	David Kelly	Aug	29
CP/Muffin	Val J. Golding	Mar	81
Direct to Page 2	Bob Halfman	Jan	57
Inverse String Printer	Stowe Keller	Sep	81
Penultimate, Penultimate Input Anything Subroutine, The	Bob Nacon	Jan	43
Quick Zapper	S. Keller/V. Golding	Oct	52
RAMfinder	Mitchel Pilot	Aug	17
Splitter	Norman L. Kushnick	Jan	53
Telecommunications with Filedump	S. Ross/V. Golding	Sep	84
Time is Now...,The	Peter D. Hipson	Aug	23

SO WHAT DID YOU EXPECT?
William K. Dolen

```
]LIST

10   POKE 33,1: FOR I
       = 39 TO 0 STEP
       - 1: POKE 32,I:
     FOR J = 1 TO 24:
     CALL  - 912: FOR
     K = 1 TO 30: NEXT
     K: NEXT J: LIST
     : NEXT I: POKE
     32,0: POKE 33,40:
     PRINT : REM SO
     WHAT DID YOU EXPECT
```


DISK DRIVES
$199

First time under $200!

At last you can afford to buy another drive!!

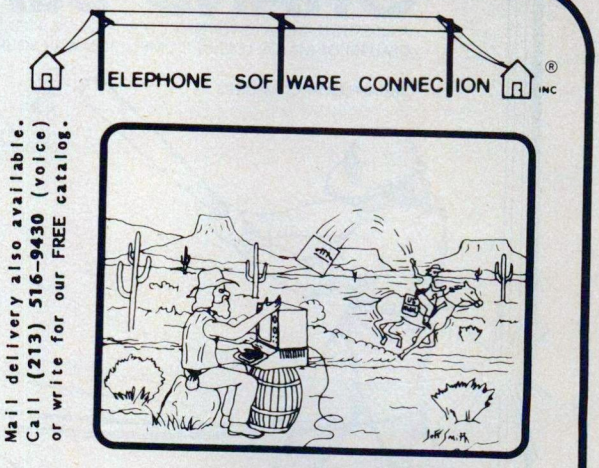
- Siemens drive mechanism (uses lead screw instead of rotating cam, for very quiet, reliable operation)
- Works with your present controller
- Can be used as 35 or 40 track drive
- Improved self-centering of diskettes
- Externally-adjustable speed control
- Handles synchronized tracks
- Manufacturer's 1-year warrantee

These superb ready-to-use drives come with standard size and color casings. *Disk controller cards available: $65.00*

Add $5 for shipping and handling. Send money order or certified check for immediate shipment, $4 extra for 2nd day delivery
Ask for free 74LS125 chip with order.

VF ASSOCIATES
6327 Western Ave, N.W., Wash., D.C. 20015
(202) 363-1313

Circle 42 on Reader Service Card



Software delivered via your MODEM.
Call us now for some FREE samples:

I N #3 #3 RETURN
CTRL A CTRL F CTRL A CTRL Q 1-213-516-9432

POST OFFICE BOX 6548 - TORRANCE, CALIFORNIA 90504

© 1983 Telephone Software Connection, Inc.

Circle 45 on Reader Service Card

DIVERSI-DOS IS THE BEST

***** NEW *** NEW *** NEW *****

TLIST — Lists BASIC files without destroying the program in memory. Use TLIST to copy lines from one program to another / Improved list format without indents, for easier editing / Visible control characters / Also lists program in memory with improved format

Insert/Delete Mode — Makes program editing a pleasure! Insert characters in the middle of a line without re-typing. Also works for data entry!

Keyboard MACROS — Enter whole phrases with a single keystroke! Make your own custom editing keys, or redefine your entire keyboard (Dvorak keyboard included)

Wildcard file names — Enter only the first few letters of a file name (searches the directory for a match)

BSAVE — "A" and "L" parameters are not necessary (uses "A" and "L" from last BLOAD)

Recognizable ESCAPE and INSERT mode cursors

Lowercase DOS commands accepted

Catalog abort key

Lists text files to screen or printer

"Of all the DOS enhancement packages reviewed in Peeling II to date, **DIVERSI-DOS** is the most powerful in terms of its capabilities coupled with its price. **DIVERSI-DOS** is the only product to speed up all areas of DOS - LOAD BLOAD, RUN BRUN, SAVE BSAVE, as well as the READ and WRITE of text files...The documentation is superb. (Rating AA)"
— Peelings II Magazine

WHAT ARE YOU WAITING FOR???

Are you tired of waiting for DOS to load and save files? Are you tired of waiting for DOS to finish so you can type again? Are you tired of waiting for your printer? When you buy **DIVERSI-DOS™**, by Bill Basham, you won't have to wait any more! Here's why:

1. DOS speed-up: Apple DOS 3.3 takes 18 disk revolutions to read a single track, whereas **DIVERSI-DOS** reads or writes a track in just 2 revolutions. This speeds up file processing tremendously (see table).
2. Keyboard Buffer: **DIVERSI-DOS** allows you to type ahead, as fast as you can, without missing a single character.
3. Print Buffer: **DIVERSI-DOS** can use a RAM card (16K-128K) to temporarily save characters before they are printed. Thus, your computer won't have to wait for your printer to finish.
4. DDMOVER: **DIVERSI-DOS** can now be moved to a RAM card to increase the available memory in a BASIC program.

DIVERSI-DOS, the QUADRUPLE utility, requires a 48K Apple II, II+ or e with DOS 3.3. A simple, menu-driven installation program is included on the un-protected disk. So what are you waiting for?

	APPLE DOS	DIVERSI-DOS
SAVE‡	27.1 sec.	5.9 sec.
LOAD‡	19.2 sec.	4.5 sec.
BSAVE*	13.6 sec.	4.1 sec.
BLOAD*	9.5 sec.	2.6 sec.
READ**	42.2 sec.	12.4 sec.
WRITE**	44.6 sec.	14.9 sec.
APPEND**	21.3 sec.	2.3 sec.

*Hi-res screen ‡ 80-sector BASIC program
** 52-sector text file

ORDER TOLL-FREE

Call NOW: 800 835-2246 ext. 127 (orders only)
For information, call 815 877-1343
Disks normally shipped within 24 hours.
Only $30: Includes 1st class or foreign airmail.
Sold by mail order only.
Return in 30 days for full refund, if not totally satisfied!

FREE — with your order — **FREE**
DOGFIGHT™ II — By Bill Basham
— A special mail-order version of the arcade game for 1 to 8 players, recently listed #6 on the best seller list!

Send $30 (U.S. funds) to:
Diversified Software Research, Inc.
5848 Crampton Court
Rockford, Illinois 61111

dsr

Name: _____

Address: _____

City _____

State: _____ Zip Code: _____

Visa Mastercard, C.O.D. or personal check accepted

Card #: _____

Exp. Date: _____

A

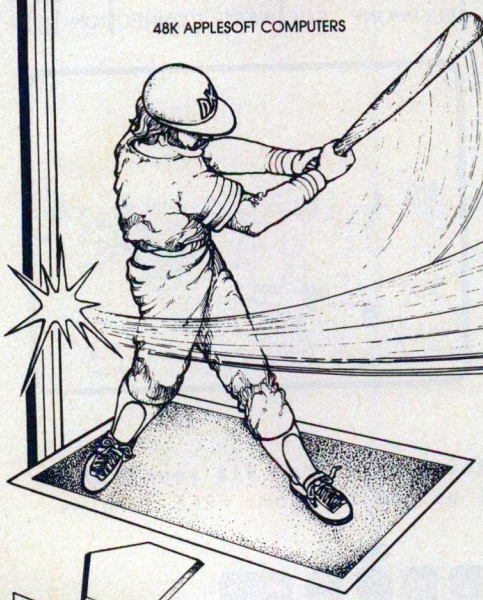
Circle 16 on Reader Service Card

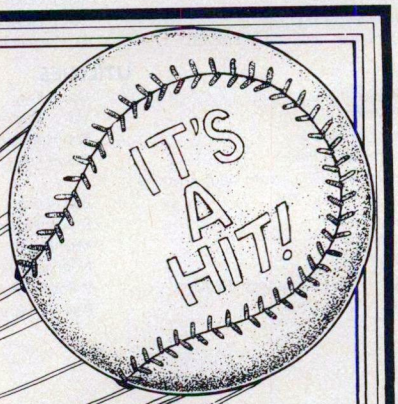
GRAND SLAM TRIVIA™

A QUESTION AND ANSWER GAME LOVINGLY CRAFTED BY BASEBALL EXPERTS TO BRING YOU THE REALISM OF MAJOR LEAGUE COMPETITION WITH YOUR FRIENDS OR COMPUTER

48K APPLESOFT COMPUTERS

ONE OF THE TACTICAL TRIVIA™ SERIES OF GAMES BY DATAKNIGHT™
YOUR SOFTWARE QUEST HAS ENDED





GRAND SLAM TRIVIA™	$34.95
ADDITIONAL QUESTION DISKS	$19.95
• WORLD SERIES • GENERAL TRIVIA	
• ENTERTAINMENT TRIVIA	

GRAND SLAM TRIVIA™

- OFFENSIVE AND DEFENSIVE STRATEGY
- 6 DIFFICULTY LEVELS FOR HANDICAPPING
- OVER 700 INTRIGUING QUESTIONS WITH HINTS
- QUALITY BLOCK GRAPHIC ANIMATION
- MULTIPLE CHOICE OR SPELLING-CHECKED ANSWERS
- NO ANNOYING RANDOM OCCURRENCES
- BOX SCORE SUMMARIES

THE COMPETITION

NO STRATEGY
2 LEVELS/NO HANDICAPPING
300 QUESTIONS/NO HINTS
STILTED SHAPES
MULTIPLE CHOICE ONLY
MANY SUCH
NONE

© 1983 EREWHON, INC.
Applesoft is a trademark of Apple Computer, Inc.

CREDIT CARD ORDERS: 215-431-8900
CHECK OR MONEY ORDER TO: DATAKNIGHT, NORMANDY CIRCLE, GLENMOORE, PA 19343

DEALER INQUIRIES INVITED

Circle 15 on Reader Service Card

A Call —A.P.P.L.E. PRODUCT REVIEW:

KOALA PAD

Alan R. Fischer/reviewer

The KoalaPad Touchtablet *$125*
Koala Technologies Corp.
3100 Patrick Henry Dr.
Santa Clara, CA 95050
(Available to A.P.P.L.E. members at $95)
Quality: Very Good
Usefulness: Graphics/Games Excellent
Price/performance: Very Good

HAVE you archived some of your older games that just aren't fun anymore. Do you aspire to produce quick and easy graphics, but don't because your joystick is too slow or uncontrollable? Have you noticed heavy breathing at the sight of a Graphics Tablet? If so, there is a product recently introduced for the Apple that revolutionizes your ability to communicate with your computer. It is called a "KoalaPad Touchtablet." If you enjoy games, expect your scores to soar! If you have a Graphics Tablet, plan on storing it in a safe place.

First we had paddles, then joysticks came along and made life easier. Those who could afford a Graphics Tablet had more sophisticated options at their disposal, however this meant more complicated software, thus there were few software packages available to take advantage of the tablet's capabilities. The introduction of the Track Ball and the Mouse made certain operations seem much easier. Another company has come out with a hi-tech light pen that allows the user to just point at the display screen to make selections or to draw. (This innovation is one of the nicest and most versatile of the developments in cursor control to-date, although the cost of the hardware is above what the average user might wish to spend for an accessory to play games with or to do casual drawing.)

By now many of you may have seen the ads for the "KoalaPad." If you have seen a demonstration of the Koala, I'm sure you'll agree that this promises to be an easy and effective way to achieve the conveniences of the earlier developments at a reasonable cost.

The KoalaPad is a product that controls the cursor on the screen thru the game port. You can draw on the High-Res or Low-Res screen or make menu selections just as you would with the joystick or game-paddles.

The Koala resembles a miniature Graphics Tablet, except that no special hardware or plug-in boards are required. The Koala's overall dimensions are approximately 6"x 8"x1" thick. The base of the unit is designed to sit at an angle or to be held in one's hand. I found both ways of use to be comfortable. The styling is sleek and functional.

The active "pad" area measures only 4½" x 4½", but don't let its size fool you, the full screen display area can be covered by the cursor. Cursor position changes are accomplished by applying pressure to any area of the "active pad surface." The active area is recessed so that one doesn't need to look at the "Pad" in order to quickly move the cursor or to find the screen corners. Watching the screen becomes natural after only a few moments of using the KoalaPad. There is also provision to fasten overlays or menu charts over the active area. The manual mentions plans for these overlays to be included with future software. A "gridwork" overlay can be made from architectural drawing Mylar and works nicely as a drawing reference.

Two ample sized buttons correspond to button (0) and button (1) as found on the standard game-paddles. These are highly responsive buttons with a "positive click" contact. Recessed into the top portion of the tablet, the buttons work very nicely, however, this location seemed awkward for some games. Perhaps having them along the sides would be more convenient for those of us who are all thumbs.

The Koala is made of a high quality, sturdy, impact resistant material. The joints and case seams are well fit. The cord that connects the pad to the game port seems a little short for two or more players standing around the computer. Another few inches would be better or you could use an expansion port.

The manufacturer recommends that only fingers or the special stylus (included with the Koala) be used on the pads' surface. Although the surface appears to be a hard plastic type of material, it has an unusual "soft" feel and it's conceivable that pens, pencils or other objects could mar or damage the sensitized surface. I prefer the stylus, however, my finger worked just as well for most operations. Drawing is much more accurate with the stylus because pressure can be accurately directed with the tip.

After only a few hours of experimenting, the stylus was showing signs of wear. As the tip wore, it stabilized much as a pencil would. Now that it has had plenty of use, it doesn't show much additional wear.

At first I noticed some difficulty with cursor response when working in the upper left corner of the screen. It seemed that control of the cursor wasn't stable and the cursor jumped around. A little practice, reduced the problem to an occasional annoyance. I traced most of the problem to the pad's sensitivity to varying pressure. To avoid the problem, one should be careful not to rest a finger on the active surface while moving the cursor

—●→

hardware

across the screen. While making High-Res drawings, you must be sure to apply continuous pressure to the sensitized surface or stray lines can occur. (They always go to the home position in the top left corner). Also be careful to release the drawing button before relaxing pressure on the pad. Random lines can also appear when using software not intended for the Koala. I found out that there is a certain amount of "noise" produced by the hardware. This tends to interfere with the accuracy of the routine that reads the value returned from the pad. The software included with the KoalaPad has filtering routines to compensate for this but most older software may not.

The values the pad sends to the computer are pre-adjusted at the factory. When using Apple paddles, the minimum/maximum values returned should be between 0 and 255 for each paddle. When using a joystick or set of paddles to poll the cursor's screen position, each corner should return a set of min/max values.

Beginning at the top left screen corner and moving clockwise, the four sets of values should be . . . 0/0, 255/0, 255/255 and 0/255. If these values cannot be generated, there may be "dead spots" on the screen.

I tried various "alignment" routines to verify that the pad was generating the proper values. I found that different programs returned different min/max values! A simple "PRINT PDL(0)" routine seemed to return the expected 0/255 values, except when in the upper-left, the values seemed to jump around and the "0" value was not reliable.

The Koala manual's notes on the Apple imply that values between 0-9 will be interpret as a null (0). Most of the programs I tested operated as expected without problems. A few games required a paddle reading of 0 to begin. These did not want to work as the lowest values I could give them was between 4 and 6. The Shot Put event of Olympic Decathlon is like that. The shot-put event returned values of 4/186, thus I could not enter the game. High-Res skiing had a problem in the ski jump event but not in the Salalom. Most other games seemed to work OK and became more enjoyable to play. Only a few of the older games showed no improvement. With most games I was able to achieve my all time best scores using the Koala.

As mentioned before, The Koala is a "Pressure sensitive cursor control pad." The cursor can be controlled from within a program, however the manual suggests a timing delay in the program read routine. External control is achieved by applying pressure to the pad area with your finger or the special stylus. Since a program routine is required to calculate a delay between reading the X and Y values, some programs that depend on an accurate reading may have problems. The simple routine,

```
10 X = PDL(0)
20 QQ = SIN(123)
30 Y = PDL(1)
40 PRINT X,Y
50 GOTO 10
```

listed in the manual returned values between 6-254.

The Koala does not provide a means to externally calibrate the value range as some joysticks do. My experiences make this feature desirable, thus any program that requires precise values might need modifications.

I decided to void my warranty and "open 'er up!" As I expected there was a potentiometer inside, however to my surprise, there is also a 2"x5" circuit board with many I.C.'s and resistors. I played with the potentiometer but couldn't get a 0 reading on either PDL(0) nor PDL(1). I did notice some value changes on the high end of the scale. The active pad area appears to be a sensitised surface sandwiched between a metal backplate and a hard/soft plastic-like external surface. After returning the setting as close as I could to where it was when I began, I decided I wasn't going to play anymore and closed the cases. I wouldn't recommend anyone to separate the housing as it will definitely void your warranty.

Manuals

Two manuals are included. One is the Touchtablet owners manual and the other is the Micro-Illustrator owners manual. Both are well written and provide adequate instruction for the first time user. The operation of the tablet and its software are so easy that little documentation is needed to master either.

Within the 12 pages of the Touchtablet manual, specific installation instructions and notes for the Apple, IBM, Atari/Commodore versions are found. Very little is said on the operation of the KoalaPad or software themselves, however simple subroutines to read the pad's values are provided.

The manufacturer states that each tablet is "tuned" for each make of computer. Upon reading the manual, I discovered that each version returns a different number range for each computer. I tried a Commodore pad on an Apple (with a commercially available adaptor) to no avail, thus a different KoalaPad really is needed for each brand of computer. The Apple version works on the Apple][,][+, and *//e*. I observed no difference in operation between a new *//e* and a 1979 Rev.1][+.

Software

Software even comes with the KoalaPad! Written by Steven Dompier, of Island Graphics and named "The Micro Illustrator," it's almost worth the purchase of the KoalaPad alone!

Menu selections are represented by little boxes on the screen. Each box has a "picture" of that selection's function in addition to a word description. Selecting menu items is as easy as pointing your finger and pressing a button. Freehand drawing, lines, circles, boxes and solids are as easy as using a Graphics Tablet! Included are a set of "cursor brushes" and a color palette of 18 "colors." Complementing options include a very nice "Fill" routine that fills one color over another and a magnify mode. These features make changing your mind easy!

—●→

hardware

The magnify mode allows Low-Res touch-ups or detail work in small areas. If you move off the "screen edge," a colored border becomes visible as an orientation aid. There is no annoying wrap around to ruin the other side of a "picture"! Selecting the help screen (?) presents a screenful of hints on the use of the various options.

Disk I/O is in standard DOS format, even though the disk is protected and uses its own "fast" DOS. I'm glad to see authors finally starting to work with the standard DOS I/O format.

The Micro-Illustrator's manual is 16 pages. The first four pages are dedicated to the introduction and booting of the system. A menu is displayed and each command is covered with a brief discussion.

Standard features include: Draw, Point, line, lines, ray, fill, frame, box, circle, disc, erase, storage, normal, magnify, help, an 18 color palette, and a set of 8 brushes. Most of these options are basic in nature and are self explanatory. Four pages are devoted to helpful hints for successful mastery in a short time (even film and shutter-speed for photographing your artwork are mentioned.).

Summary

While trying the KoalaPad with some older software, I discovered a whole new dimension to each program. In most cases the cursor response was much more controllable. Most programs were more enjoyable or easier to use.

Some of the software I tested on the Koala included:

The Complete Graphics System
Paddle Graphics
Horizon V
Olympic Decathalon
Thief
Zaxon
Castle Wolfenstien
Gran Prix
Pegasus
Choplifter
High-Res Skiing
Tailgunner
Missile Command
Tranquility Base
Pinball Construction Set

Olympic Decathlon, High-Res Skiing, and Font-trix showed problems. The Complete Graphics System was more awkward than the Micro-Illustrator and although it offers additional options, you must be careful of cursor jump and stray lines.

Although I experienced some difficulties, the KoalaPad is in my opinion, a noteworthy development and well worth the asking price. The Micro Illustrator is a very versatile drawing program and a good combination with the Koala. Both children and adults will find their computer to be more fun with a Koala attached.

hardware

Introducing the New Locksmith 5.0™

The World's Finest Nibble-Copier for Apple® Protected Software **Just Got Better!**

When you invest in software that's vital to your everyday usage you can't afford to be without a backup. And nobody can do it better for you than Locksmith. (Our earlier model 4.0 was recently rated by Apple users as their 3rd most favorite software program.)

Locksmith 5.0 backs up virtually all the latest Apple disks—even those with the most recent protection devices. And, thanks to the new interactive "conversational" program language, you can upgrade this Locksmith to decipher new protection devices as they appear. You keep Locksmith 5.0 **state-of-the-art!**

You also get all the utilities that made Locksmith so popular...and more.
• Media Surface Check • Disk-drive Speed Calibration • Degauss and Erase • Nibble-Editor • Quickscan • plus many others.

Special Trade-In Offer for Owners of Locksmith 3. and 4. Programs

Upgrade your current Locksmith to 5.0 at a substantial savings over the purchase of a new 5.0. Call or write for details.

A New Manual

Clear and easy-to-understand, this newly written manual will take you step-by-step through all the Locksmith options. Instructions are precise and detailed.

Locksmith 5.0 only $99.95 plus $3.50 for shipping and handling.

Call Toll-Free 1-800-238-2300
(In Kansas 1-800-362-2421)

Omega MicroWare
222 S. Riverside Plaza
Chicago, IL 60606
(312) 648-4844

Locksmith and Omega MicroWare are trademarks of Omega MicroWare, Inc. Apple is a registered trademark of Apple Computer, Inc.

© 1983 Omega MicroWare, Inc.

Circle 21 on Reader Service Card

THE GAMING ROOM

Val J. Golding

HOW about another appeal to our readers for a kicker, and perhaps a word or three about this column. **The Gaming Room** started when we felt that games and gaming were being overlooked in *Call -A.P.P.L.E.* We are entirely open to suggestions from our readers as to what you would like to see here. Our present "configuration" is to present a goodly share of game reviews, comments about current games, hints on winning and excerpts from readers' letters.

As we find complete articles on gaming, they will be published under the column title, i.e., "Out Wizarding Wizardry" last month. We invite our readers to let us know what they want to see and to also contribute mini-reviews of new games that would be of interest. After all, the more material we get, the less we have to write!

In past issues we have reviewed at most two or three games, and while the reviews were fairly comprehensive, there were always more games to review than we had space for, therefore we'll try a new approach this time and review a few more games in less detail. Let us know if this is what you want.

A pair of games have come our way from *Weekly Reader Family Software* (a division of Xerox Educational Publications), 245 Long Hill Road, Middletown, CT 06457. This is a new name to us, so these are likely among their first releases. If **Fat City** and **Chivalry** are any indication, we have a new line of quality software available.

Priced at $39.95 and $49.95 respectively, our first impression was that they were overpriced. However, on a second glance, and looking at the time and effort devoted to the first class packaging, we feel that although somewhat high, they are fairly priced. As expected, both programs are copy protected, although the enclosed warranty cards do indicate free replacement for defective diskettes, which ain't too shabby.

Fat City comes in a stiff plastic folder which contains the program diskette, eight decals, a neat poster, warranty registration card and a durable 16-page documentation booklet, printed in large, easy to read type and many illustrations. Unlike much game documentation, nothing is left to the users imagination.

The player is the operator of a large wrecking crane, and his/her goal is to demolish ten cities slated for urban renewal. Of course, rats are always the last to leave a doomed building, and they present the only hazards to the player by tossing bricks and other debris at the cab of the wrecking crane. Each hit costs the crane fuel as it attempts to dodge. If all buildings are successfully demolished, the player receives bonus points and moves on to the next city.

Scoring is set up for each city (round) and you have an opportunity at any time to go back and improve your score on a prior round. In four evenings of play, we found our way (just barely)to round eight and a score of 75,000+. As with most games, experience is the key to winning; as you become familiar with the most effective techniques, it becomes easier and easier to progress to the next round. This is a game that we believe can present a challenge and hold your interest over a long period. We know we will be playing it for some time to come.

Chivalry is something just a wee bit different. It is a well engineered meld of adventure game, board game and arcade game. It comes in a deluxe hard box package which includes four game pieces (up to four players can play), a playing board and 16 page players manual which describes each of the 20 or so sub-games a player may encounter on his/her way to rescue the King from the Black Knight. Some of the sub-games are so simple you might think "This is no challenge," and then you get to the castle of the Black Knight. Suffice to say that after three hours of play, our rescue mission was unsuccessful.

Both games are well designed and bug-free with exceptional graphics. Our only two gripes concern the excruciatingly long idle period on Fat City and the mindless, out-of-tune music on both programs. Surely in this stage of Apple development, more sophisticated music could be played. Manufacturers please take note: How about adding a method of turning off the music, but not the rest of the sound. Or better yet, some good two-voice, in tune melodies.

From *Sir-Tech Software, Inc.,* 6 Main St., Ogdensburg, NY 13669, comes a new **Wizardry** scenario: **Legacy of Llygamyn,** the third in the series. Wizardry is one of the oldest and most famous adventure games around, and has appeared in the number one spot of *Softalk's* best-seller list many times. "Legacy" requires characters of any experience level created with **Proving Grounds of the Mad Overlord** and introduces an entirely new concept of windowed graphics display in 3-D. In addition to the 3-D maze display, Legacy offers up to six levels of mazes and can be played by up to six players. Although *Call -A.P.P.L.E.* has previously printed two articles on Wizardry, we have never (hint-hint) run a comprehensive review. We would look forward to publishing one.

Penguin Software, 830 Fourth Ave., Geneva, IL 60134, has come up with a sequel to **The Spy's Demise,** entitled **The Spy Strikes Back.** The familiar spy returns in the setting of a German fortress. In the spy's new environs, there are five floors, each divided into 24 sections of 16 rooms each, more than enough to occupy one for aeons. A timer ticks away, reducing one by one, the points you will receive when you find your way out. Should you be spotted by guards, the points you receive are cut in half. As in the previous "Spy," clues abound in each room, there are clues to be gathered which for the astute few, can lead to $100 software prizes from Penguin. With a $19.95 price tag, "Strikes Back" is a software bargain.

There's more to report, but we've run out of space. Once more, readers, let's hear from you!

PRODOS AND FRIENDS

Cecil Fretwell

WITHOUT regard to whose computer system it is, giving a hacker like me a new operating system is like giving a kid a new toy. As soon as I heard that Apple would soon release a new operating system called ProDOS, I immediately doled out the proper amount of cash and signed the necessary agreements to increase my licensed development efforts.

ProDOS stands for Professional DOS, a name which is very indicative because the system is truly professional, yet friendly to the not so experienced user. Having some 25 years experience in larger computer systems, I am very impressed with ProDOS. Charpentier and Sparks did an excellent job of introducing ProDOS, therefore, I will try to be the first to start a series of technical notes on ProDOS.

My developer's disk shows a date of 1-SEP-83 and comes up with the following screen when the system is booted.

If your system does not have extended memory, the screen will show 64K instead of 128K. My extended memory is the one offered by A.P.P.L.E. and works great.

To see how all this magical information came about, do a catalog of the disk to show

```
****************************************
*                                      *
*        PRODOS DEVELOPER'S DISK       *
*                                      *
* COPYRIGHT APPLE COMPUTER, INC. 1983  *
*                                      *
****************************************
               STARTUP DISK: /PRODOS/
YOUR Apple //e HAS:
     128K OF RANDOM ACCESS MEMORY
     AppleSOFT IN ROM
     SLOT 1: I/O CARD
     SLOT 2: EMPTY
     SLOT 3: 80-COLUMN CARD
     SLOT 4: EMPTY
     SLOT 5: EMPTY
     SLOT 6: DISK DRIVE
     SLOT 7: EMPTY
```

```
]CAT
/PRODOS
NAME              TYPE  BLOCKS  MODIFIED
*PRODOS           SYS       29   1-SEP-83
*BASIC.SYSTEM     SYS       21   1-SEP-83
*CONVERT          SYS       42   1-SEP-83
*FILER            SYS       51   1-SEP-83
 EXERCISER        SYS       16  29-JUL-83
 STARTUP          BAS        7   1-AUG-83
*BASIC.RUNTIME    SYS       21   1-SEP-83
 EDASM.SYSTEM     SYS        9   1-AUG-83
 EDASM.ED         BIN       17  14-JUN-83
 EDASM.ASM        BIN       29  14-JUN-83
 FORMATTER        TXT       15   4-AUG-83
 BUGBYTER         BIN       15   4-APR-83
BLOCKS FREE:    1      BLOCKS USED:  279
```

or

```
]CATALOG
/PRODOS
NAME              TYPE  BLOCKS  MODIFIED         CREATED          ENDFILE SUBTYPE
*PRODOS           SYS       29   1-SEP-83  0:00   1-SEP-83 15:14    14336
*BASIC.SYSTEM     SYS       21   1-SEP-83  0:00   1-SEP-83 15:13    10240
*CONVERT          SYS       42   1-SEP-83  0:00   1-SEP-83 15:15    20481
*FILER            SYS       51   1-SEP-83  0:00   1-SEP-83 15:14    25344
 EXERCISER        SYS       16  29-JUL-83 14:31   1-AUG-83 10:12     7362
 STARTUP          BAS        7   1-AUG-83 13:44   1-AUG-83 10:12     2762
*BASIC.RUNTIME    SYS       21   1-SEP-83  0:00   1-SEP-83 15:13    10240
 EDASM.SYSTEM     SYS        9   1-AUG-83 11:36   1-AUG-83 10:15     4095
 EDASM.ED         BIN       17  14-JUN-83  9:50   1-AUG-83 10:15     7935 A=$3000
 EDASM.ASM        BIN       29  14-JUN-83 10:17   1-AUG-83 10:15    14079 A=$3000
 FORMATTER        TXT       15   4-AUG-83  8:03   2-SEP-83  9:45     6733 R=    0
 BUGBYTER         BIN       15   4-APR-83  0:00   1-AUG-83 10:37     6944 A=$2000
BLOCKS FREE:    1      BLOCKS USED:  279      TOTAL BLOCKS:  280
```


technical

PRODOS AND FRIENDS

Time out for short note on the catalog process. Right away I liked the idea of being able to date and time stamp catalog entries. Even if you do not have a clock such as ThunderClock, Sparks' version of ProDOS has a BASIC program which will let you set the date permanently and you can change the time manually. Don't let the blocks business fool you. DOS 3.3 used 256 byte sectors and ProDOS uses 512 byte blocks; you still have about the same amount of disk space available.

I also like the CREATED and MODIFIED idea. My only question is how could EDASM.ED be created 1-AUG-83 and modified 14-JUN-83. Without further digging, Apple may be using the MODIFIED area to indicate the latest release of the module and may be using the CREATED area to indicate when the copy was made. If that is not the case, this could be a bug. Bugs are to be expected in any new release of a system whether it be IBM or DEC or anyone else.

Another excellent feature of the long catalog form is the ENDFILE and SUBTYPE columns. So far I have found the BIN file information to be quite valuable. Not crawling before I walked, after I played with ProDOS for a few evenings, I launched a two week project of converting Glen Bredon's Big Mac 48K to ProDOS. In that code, he saves the object code using a BSAVE then modifies the first sector of the object file to reflect the proper starting address of the object code. That took a lot of knowledge of DOS 3.3, under ProDOS it was a breeze after I understood what was going on. More on this later.

The last thing I like about the catalog process is the automatic free space information. Many of us have modified DOS 3.3 to obtain this information.

Back to the magic of the first screen. At boot time, the first module brought into memory is the file called PRODOS. This guy is the kernel (nucleus or core is one of Webster's definitions) for all systems. He is loaded into $BF00-$BFFF and the language card. This provides what Apple calls the Machine Language Interface (MLI) which represents all of the disk I/O for the system, among other things. More on this later. As far as I can tell, a copy of $F800-$FFFF in ROM to the language card is not performed, therefore one must assume that no special ROM Monitor is used by the MLI or BASIC.SYSTEM.

Having brought in PRODOS, the boot process then loads the first file of type SYS into memory. For the disk used as a basis of discussion for this article, this is BASIC.SYSTEM. The hackers are going to love this idea because they can put together a system without having to write extensive I/O code. As an experiment, I formatted a disk, copied PRODOS to it, and then developed a program which did nothing more than jump to the ROM monitor. This was a tough program! It only had JMP $FF69 at the required location $2000. SYS type programs must be loaded into $2000, after which they can be moved to any other desired area. Calling my system JUNK.SYSTEM, my disk had

PRODOS

JUNK.SYSTEM

It booted and came up with the * prompt. I had the monitor and all its goodies. The system could not do much because I was limited to monitor commands and had to be careful not to use commands that transferred control to assumed code such as BASIC.SYSTEM.

Back to the discussion of the boot process, at this point, BASIC.SYSTEM is in control. It loads and executes the BASIC program called STARTUP. This is the old HELLO concept from DOS 3.3. Although one must call it STARTUP, the system is perfectly happy with the fact that the file may also be an EXEC file or a BIN file representing assembly code.

STARTUP can be anything your heart desires. It does not even have to be present on disk or it could be as simple as

```
10 END
```

with the end result of the] prompt. At this point, the new user familiar with DOS 3.3 and Applesoft Basic is back into totally familiar territory. The system will even accept lower case commands such as catalog, therefore, the disposition of the CAPS LOCK key on the *IIe* is no longer quite the "problem" as it was under DOS 3.3. Any command that is passed to Applesoft Basic, e.g., LIST, must be entered in capital letters. Charpentier and Sparks point out, as long as one does not use weird PEEKs or POKEs or call specialized assembly programs from BASIC, *nothing has changed.* If anything the system is better for its improved garbage collection alone.

The screen shown at the beginning of this article is the result of a STARTUP program written in BASIC. When the whole boot process finishes, PRODOS has placed a lot of valuable information in what is known as the System Global Page. This resides in $BF00-$BFFF and is well documented by Apple. To illustrate the information available, $BF98 tells you what type of Apple you have,][,][+, *IIe*, etc., how much memory the system has, whether an 80 column card is present, and whether the system has a ThunderClock.

Hackers be patient for a little while longer. Two more files on the disk need some attention. You have some files to copy over to ProDOS? No problem. With the] prompt present, use the new RUN command and enter

]-CONVERT

Very shortly (due to ProDOS's increased speed), a menu will appear.

technical

In fact all utilities are menu driven and user friendly. In a matter of seconds, if you have two drives, you can copy a DOS 3.3 file to ProDOS format. Now you say you have several files to copy and dread the thought of entering all those file names from the keyboard. Not to worry. Via the transfer command, you can first obtain a catalog of the DOS 3.3 disk. In the resulting display, you use the arrow keys to move the cursor to the desired file name then press the space bar. A little "tick mark" is placed there and you then use the arrow keys to move to the next file name. When you have all the desired files "tick-marked," you press RETURN and CONVERT copies of all the files you specified. He is even a nice guy and stops when he reaches one that already exists on the ProDOS disk and lets you decide whether or not to replace the existing file.

By the way, the pathname concept under ProDOS is great. It greatly resembles the Pascal method of operation. The only problem I have is breaking the habit of imbedding spaces in a file name (excuse please, pathname). If the DOS 3.3 file has spaces in it, CONVERT converts these to periods. Another "problem" is that file names within a pathname are limited to 15 characters so one has to be a little more inventive when one has been used to the 30 characters allowed in a DOS 3.3 name. CONVERT truncates a name to 15 characters, so in some cases, one can end up with duplicate name problems.

No need to re-boot when done with CONVERT. When he exits, he prompts you for the system you want to activate next. Thus you can return to BASIC.SYSTEM or switch to FILER.

In essence, FILER is mainly FID under DOS 3.3, just as CONVERT replaces MUFFIN, and provides the INIT command removed from DOS 3.3. He is menu driven and very friendly. There are two added features I like. One of these is the ability to check out a disk for bad sectors, which takes only a few seconds. Another feature is the ability to compare two files. I used this to compare my GPLE disk to a friend's GPLE disk. In a matter of seconds I was able to tell him that his older GPLE was the same as my newer GPLE, and that his suspected problem was probably a "short circuit between the ears" or a bug in GPLE.

The only utility missing is the RENUMBER utility available on the DOS 3.3 disk. In talking with the technical support people at Apple, it appears that they will provide RENUMBER on a "tool kit" disk. Until Apple comes up with one, we will have to convert a ProDOS BASIC program to DOS 3.3 using CONVERT, renumber it with RENUMBER, PLE, etc. then CONVERT it back to ProDOS.

Now you hackers and other experienced types, let's have some fun. I will be mildly surprised if someone finds it necessary to modify the MLI or BASIC.SYSTEM similar to the modifications various people have made to DOS 3.3. The interface to the MLI is extremely simple in principle. It looks something like

```
JSR   MLI
DB    CMDNUM
DW    CMDLIST
BCS   ERROR
```

Apple says that MLI will always be located at $BF00 and we could really care less if the JMP at $BF00 changes from one release to another. CMDNUM is a code number for over 25 documented functions that can be performed by the MLI. One can create files, open files, write to files, obtain information about files, allocate interrupt vectors, etc. I have not begun to dig into what all of these functions do, but feel that every conceivable situation has been covered.

Each command has a command list, CMDLIST, associated with it. As a safeguard, one must start that list with a byte that contains the count of the required parameters for the list. Put a number here other than what the book specifies and you get an error.

Upon return from the MLI to the location following CMDLIST, one can determine whether an error has occurred in the process. Even though the pre-release manuals use a BNE in their code example to branch to error handling, this may be incorrect from my own personal experience. The description following the example in the manual states that the carry bit will be clear if no error occurred or the carry bit will be set and A will contain an error code.

As a simple example, suppose one wants the current prefix for the system. The code would be

```
        JSR   MLI
        DB    $C7
        DW    PREFIX
        BCS   PREERROR
        .....
        .....
PREFIX  DB    $1
        DW    NAME
        .....
        .....
NAME    DS    128
```

A successful return places the current prefix in the buffer called NAME in this case. The first byte of the buffer is the length of the prefix name and the remainder of the buffer contains the prefix name in ASCII. The manual says this is assumed to be a 128 byte buffer. It appears that it need only be as long as necessary to hold the prefix name plus the length byte.

I used this as one of the MLI examples because it nailed me for several hours during my Big Mac 48K conversion efforts. In this effort, I booted with BASIC.SYSTEM active then BRUN my converted Big Mac, I called it PRODOS.TINY.MAC. Part of Big Mac's menu is used to show the active drive. Under ProDOS, prefix is more applicable. When I used MLI to get the prefix, it returned with a length of zero. If one uses the PREFIX command in BASIC to explicitly specify the prefix even if it represents the volume booted from, then the prefix name is returned. I used the following simple BASIC program to set the prefix to the boot volume name and then fire up PRODOS.TINY.MAC.

```
10  PRINT CHR$(4);"PREFIX":
    INPUT PR$:PRINT
    CHR$(4);"PREFIX";PR$
20  PRINT CHR$(4);"BRUN
    PRODOS.TINY.MAC"
```

Or one can use the MLI call that lets you change the prefix, in which case, a prefix will be returned to you if you ask for it. Or one may use other MLI calls to obtain a volume name to be used as a prefix.

technical

This simple use of the MLI taught me another lesson in a hurry. Part of that lesson was learned when processing pathnames. Big Mac 48K uses the stack area in $100 to maintain the file name loaded. This is sort of dangerous, but usually safe if you leave plenty of room for a stack at the end of $100. If you don't leave enough room the stack will grow into your valuable data area and probably produce weird results.

Copying Glen's idea, I specified a small unused area of $100 in which to return the prefix so I could display it. I kept getting a return code of $56 with carry set. The book says this represents a bad buffer address. I fought this and my lack of MLI experience for a couple of evenings before I decided it was time to learn how to use EXERCISER, another SYS file on the disk.

EXERCISER is a system that lets you play with about all of the functions available under MLI. I used it to look at the $100 area and got an indication that MLI regards this as a protected area. The sun rose in bright colors at midnight because I then remembered that when ProDOS and friends are called, they keep idiots like me from specifying buffers that are regarded as private property. The manuals are very explicit about this business of protected memory. ProDOS and friends can't help it if I do something like an STA into their private property which will probably foul them up (voice of experience speaking again). ProDOS protects the area $100–$1FF, $400–$7FF, $BF00–$BFFF, and $D000–$FFFF.

Let's leave EXERCISER for a minute and dig into another problem I had. Big Mac 48K assembles object code into an area starting at $8000. When one uses the SAVE command to store the object code on disk, Glen first does a BSAVE using $8000 as an address and, of course, the proper length of the code. Having extensive knowledge of DOS 3.3, he then used RWTS to eventually re-read the first sector of the stored code, changed its first two bytes to show the proper address, and wrote the sector back to disk.

The MLI makes this process very simple with about the same amount of code. One first uses the GET FILE INFO function to read in a pathname's attributes. Since the object code is a BIN, binary object code, file, bytes 5 and 6 of the buffer used for the MLI call show the load address for the file. Changing these two bytes to the proper load address for the actual code and performing a SET FILE INFO changed the file with the end result that it can be BLOADed properly.

In testing the FILE INFO calls to MLI, I kept getting the $40 return code which is an invalid pathname syntax error. I had followed all the rules of specifying a pathname buffer that started with a length count followed by the desired pathname. I did not start the name with a /, so I assumed the system would pick up the proper prefix. Until I solved the PREFIX problem described earlier, this bug stayed around for several evenings.

Enough on the MLI for now. I am sure that several good articles will be developed around the MLI. Since I wanted to use BASIC.SYSTEM to help me out with the BLOADs and BSAVEs in Big Mac just as Glen does at the assembly language level, I sent a buffer through COUT which started with $84 (CHR$(4)) and ended with the required carriage return, $8D. This worked like a champ under DOS 3.3 but doesn't work using BASIC.SYSTEM. I would like to think I made an independent discovery of this fact, however, I later received a technical note from Apple confirming this fact.

So the next step was to start looking in the manuals under the heading of The BASIC System Program's Global Page. This occupies $BE00–$BFFF and is well documented by Apple. Lo and behold, Section A.3.1 told me that if I placed a DOS command starting at $200 and ending with a carriage return then did a JSR $BE03, BASIC.SYSTEM would perform the command as if I had entered it from the keyboard. Upon return from this subroutine, they told me that if $BE0F is zero, the command was successful, if not successful, then load A with the contents of $BE0F and call PRINTERR at $BE0C to have an error message displayed.

Everything seemed to be "hunky dory" until I told my Big Mac to LOAD a source file that did not exist on disk. The stupid thing came back with success and placed pure garbage in the source area. It behaved itself under the] prompt and even the EXERCISER behaved itself.

Now was the time to learn how BUGBYTER could help me. I understand that Apple may sell this software and other software on a "tool kit" disk. I did not try to use my old friend SYMBOL SYMON written by Mike Johnson because I already knew that most old DOS 3.3 assembly language programs do not migrate to ProDOS. Also, SYMBOL SYMON must be loaded into a fixed area of memory, BUGBYTER does not because he relocates himself to the address area you specify.

Once you get the hang of BUGBYTER and turn his buzzer off before the wife complains of a headache from it, BUGBYTER is a powerful debugging tool. He lets you set breakpoints, examine memory, single step the program, look at the registers, etc. So he helped me with the problem in reading a non-existent file.

I soon found that if one passes a command through $BE03 that upon return the result is the same as for a MLI call. If carry is clear, the process was successful. If carry is set, A will contain the error code and PRINTERR can then be used for reporting the error. By the way, if you call PRINTERR with a code that DOS does not understand, he doesn't care. You will just get some sort of weird message which at least lets you know something drastic happened.

technical

As if I did not have enough problems, another subtle one made the conversion process a two week effort. Under DOS 3.3, you could use the keyboard area $200 area safely without DOS messing it up and as long as your keystroke efforts did not wipe out a valuable data area. ProDOS users beware. When BASIC.SYSTEM parses a pathname, he generates the full pathname using the $280 area. He also uses the start of $200 to somehow store the prefix temporarily. Apple documented the zero page usage well in the manuals and told us in a technical note to leave $200-$2FF alone.

So much for BASIC.SYSTEM. An earlier version of the developer disk contained a file called BASIC.RUNTIME, which is a SYS file. It is shown on the example directory listings I have used, but has been subsequently removed by Apple.

The developer diskette also shows the files EDASM.SYSTEM, EDASM.ED, and EDASM.ASM. This editor/assembler system which closely resembles Big Mac, is based on Apple's DOS Toolkit ED/ASM, and while it squashes some bugs, new ones are also introduced. I have had very little chance to use it in depth. However, based on the tutorials and the manuals, it provides for list files on disk and relocatable object code if one so specifies. I am a firm believer in cross reference listings, therefore, I hope Apple will consider providing such a utility in the future.

One will note that the developer disk had one block free. How can the single disk user store files on a disk without having to play musical disks? Simply format a disk and copy PRODOS, BASIC.SYSTEM, and STARTUP to the new disk. You now have over 200 blocks to store programs. I tried this and it works. Add FILER and CONVERT to the disk and you have over 100 blocks free.

For the hacker who does a lot of assembly language stuff, create a disk with PRODOS, EDASM.SYSTEM, EDASM.ED, EDASM.ASM, and BUGBYTER. You have over 180 blocks to work with. The point is one can configure a disk to suit their needs.

The last file on the developer disk is FORMATTER. It is source code for the assembler to allow one to write their own system to include format disk capability. A technical note from Apple explains how to use FORMATTER.

In my opinion, ProDOS has a great future. It is up to us as users and developers to make it as powerful as Apple intends it to be.

technical

The $175 disk drive for Apple II

Now, you can have a 5-¼" floppy disk drive compatible with the Apple II for the unbelievably low price of $175. This includes the drive, cabinet, cable and our standard warranty including 90 days parts and labor. For the best deal in town, call StarLogic today.

TELEPHONE ORDERS ONLY
MASTERCARD, VISA, CASHIER'S CHECKS COD ORDERS ACCEPTED
(213) 883-0587

StarLogic

20860 Plummer Street
Chatsworth, CA 91311

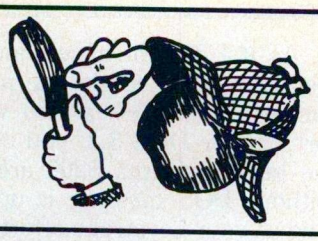
Apple is a registered trademark of Apple Computer, Inc.

(Also compatible with Franklin ACE)

Circle 44 on Reader Service Card

NORTHERN SPY:

A CANADIAN HARVEST



Rick Sutcliffe/Call —A.P.P.L.E. staff writer

SUMMER is almost over at this writing, and what a fine season it has been. Everything is early this year, some parts of the garden ready to harvest as much as a month before normal. Just last night we picked our crop of Kings, and the Gravensteins seem not far behind. Of course the Spys will be much later even this time, but they are worth waiting for. So, let's do a little Fall roundup in this space, and see what the month's harvest brings us from the electronic garden. (Oh, I know — you'll read this in the dead of Winter — but why not?)

What's up, eh?

Some of you have been wondering just how different things are in computerdom up here in the magic land of the North. How do we get electricity to our igloos; can an Apple survive -40C; do we have flush toilets; how do we type with our mittens on, etc. If my comments above do not settle your fears on some of these scores, let me assure you that computer people in Canada put their shoes on in the same way as they do everywhere — both at once and into their mouths in most cases.

The flavour of things is a little different though. (Not the shoes, silly, I'm going to be serious for a while.) Consider the magazines which are available here. Besides the usual large U.S. publications, some of which can be obtained worldwide, we have a few of our own, each with its own particular constituency. Some are not quite what you would expect to see South of the border. (There's a phrase we use a lot, which you might not appreciate.)

A Nuts and Bolts Paper

Take the new *Computing Now!*, out of Toronto, a spinoff of *Electronics Today International* (ETI for short). The latter had in recent months become quite computer oriented, so a move like this was inevitable. At this early stage (issue #4 in July 83) the two share virtually all the same staff, and are similar in style and advertiser list.

And those ads do tell one a lot, betraying as they do the origins of CN! in the Electronics world. Most of them originate with the seemingly endless parade of Eastern Canadian manufacturers and distributors of kits, components, bare boards, and assembled versions of Apple workalikes and their peripherals. A typical motherboard offered today is made in Canada and offers a wide range of enhancements over the original Apple. Some features offered include 80 column, Z-80, and disk controller, with anywhere from 48K to 192K of RAM — and all of this on board!

Although fully assembled and ROM populated el-cheapo clones also abound, such boards are usually sold either bare or in kit form with blank EPROMS. Programmers for the latter (complete with software to operate same) can be had for as little as $79 from companies like Exeltronics. (All addresses below).

It is my understanding that anyone can make a copy of the Apple ROM code provided two conditions are met:

1) That person already owns a legitimate (purchased from Apple) copy of the code in question, either on a system master disk, or in ROMs that have been purchased apart from a computer, and
2) The copy once made is retained for the sole and exclusive use of the maker and is not subsequently sold or given away to some other party.

I should note that the better "Apple" boards available have been extensively re-designed and some bear little physical resemblance to the original. Thus they would not appear to violate any patent restrictions. The addresses of two typical distributors are below.

Grokking with Another Steve

Getting back to the magazine itself, the editorial content of CN! is eclectic, covering software and hardware for the whole industry, despite the preponderance of Apple related ads. At this point, the bulk of the material is written by the editor, Steve Rimmer, and his prose may take some getting used to. Some examples:

"The documentation seems to be a bit lacking in some of the grotty details."

Speaking of a do-it-yourself EPROM eraser: "This device also serves the handy secondary function of being a first rate roach killer."

And in another product review: "...certain really uncool things about it... (but) its overall interesting karma in some way makes up for this."

Ah yes — well they need writers, and the thing seems to be off to a good start. This may just be your paper if you are into the hardware side of computing and like to read totally awesome mags that send you right out of your space.

For the Pinstripers

A somewhat more business oriented publication is *InfoAge,* out of Willowdale, and now into its second year. This one is a little more high toned, but does serve the same micro systems market, has some educational material, and offers a column on the Apple. It might be worth a look, as it seems to be improving exponentially. I might even renew.

Go West Young Writer

Typical of a new information vehicle is the large format *Compuwest* out of Vancouver. This one is a regional publication and carries the subtitle "The Western Computing Magazine." Actually, it has been just a Vancouver paper so far, and a giveaway at that, but it does show some promise. Its first few issues were retail trade oriented, but owner David Jamieson tells me that a shift to the business realm should be expected with the September issue.

An advantage to these regionals, which are springing up all over the place, is the immediacy of the content. You get ads and articles that do not have time to become out of date by mouldering on an Editor's desk for six months to a year before they see the light of day.

This one has advertising support from Apple Canada, and I may just send them a short piece or two, to brighten their day, or whatever. Actually, I am looking for an outlet for a new column, which will cover industry-wide issues rather than being Apple specific. Seems I have far too much material for this space alone.

More Clones

As you might have guessed from my earlier comments and columns, we are inundated with clones here in Canada. I would estimate that a new "brand name" turns up at the rate of about two a week. Stores will even import these things, and replace the name plate with one of their own. What has been stolen once can be swiped again. The law (and Apple Canada) seem powerless to do anything about the situation here, so there have been no Customs seizures in this country as there have been South of the border.

One which you may be hearing more about in the next little while is called the Circle-64. Imported to the West coast by Robert Strachan of Vancouver, it boasts 64K of on-board RAM, a numeric keypad, and two "FUNCTION" keys for access to keyboard macros. Apparently, the plan is to import it to the States with a single boot ROM (there is only one ROM socket on the board) and a "Softcard" installed — telling Customs that it is a CP/M machine (which it will be at that point).

The local wholesalers will then remove the ROM and replace it with a piggyback board holding three 2732's which of course contain the Apple ROM code. They get their Apple compatibility and avoid the customs hassle at the same time.

Perhaps they will get away with it, but as I've said again and again, the sale of somebody else's code is plain theft — even if the law is unable to stop it. Curiously, some businessmen I have talked to do not even realize that the machine they are selling contains any firmware; they merely cite hardware design changes and the fact that "everybody is doing it" to support their claim that they are into nothing morally or legally wrong.

(Note — added in proof — As of September 1, the pirating of ROM code was declared by the Philadelphia court of appeals to be a violation of U.S. Copyright law. Franklin, which has now lost its case, will no doubt appeal to the Supreme Court, but Apple would seem to have grounds for an injunction and damages if the decision is upheld. Due to differences in the copyright laws, a similar decision in Canada seems unlikely at this time. — R.S.)

Strachan at least seemed aware of the existence of a possible ethical problem, but rationalized it by citing what he called "Apple's ripoff pricing structure." In his view, he was providing a much needed service to the consumer, and this outweighed any potential problems with copyright or patent law, which he assumes (apparently correctly) cannot be applied in Canada.

But Apple invented the thing and they advertise and support it world wide. These small dealers can never duplicate that support and pay those expenses without their own prices rising to similar levels.

Their best answer back is another big advance on the *IIe* (Z-80, 80-column, and 256K on board) and/or a price reduction to the $700 range. At that point, the clones will all become non-competitive and drop out of sight. It can be done folks, with sufficiently high volume, and a product line that includes the $1250 Mac to push the price slot of the *IIe* down.

How does the Circle compare with last month's fruit basket? The firmware is the usual 99% copy, so compatibility is no problem for most things. The patch to handle lower case is clean and short, and starts at $FBB3, so a program can tell if it is in a Circle by looking for the unique "signature byte" value of $C9 at this location. In this way, it can be software distinguished from a plus, nonplus, e, Franklin, etc. each of which have different values at this address.

Such an ability is probably important in this case because lower case is achieved by pressing the shift key — a direct violation of all that goes by the name of common sense. If you do not depress the shift key, you get upper case. Yikes!

On the positive side are the keypad, extra +,-,/,*, decimal, pause and break and those two FUNC keys — one on the left side near CNTRL and one on the right beside the spacebar. When depressed along with any other key on the board, they produce a macro (typed word). For instance, <FUNC ?> yields CATALOG. Other keys produce various DOS and BASIC keywords, and in conjunction with the SHFT, CNTRL, AND SHFT/CNTRL keys will also produce reserved words from FORTH, Pascal, or CP/M.

This can be a good feature, but as in most keyboards of this type, the timing of the macro production is correct only for standard environments, as their letters are produced on the screen very rapidly. If you type the above example while in Visicalc, you gat CAG, and if you do it from Screenwriter, only CG. The letters are produced too fast for the rather lengthy output routines of these programs to capture them.

Construction is solid, all units being tested in Taiwan before shipping, and the five Amp power supply contains a built in fan (a bit noisy). The mother board uses 4164's so some chips are saved. It retails for CDN$879 without CP/M card, or an extra $70 with.

On the negative side is that weird shift key arrangement, the lack of an extra <RETURN> on the keypad, and the non <CNTRL> - protected Reset key. (It is recessed and relocated though.) The keyboard rates a five — ahead of the old Apple, but behind the *IIe*, and well behind the Franklin.

So is the overall rating behind the Franklin in the "improved copy" category — though with a better keyboard and proper shift arrangement it would rate higher. Here is a short chart:

Circle64	
Innovativeness	**4**
Faithfulness to][+	**8.5**
Compatible to *IIe*	**4.5**
Physical Considerations Overall	**6.3**
Reliability	**8**
Keyboard Overall	**7**
GRAND TOTAL RATING	**6.4**

Again, a word about the accompanying disk drive. I was supplied with a no-brand-name half height drive of dubious Taiwanese ancestry. It had an auto eject mechanism, but otherwise worked as an Apple drive would. At first, I found it to be very slow, and unable to find track zero on a cold boot without manual help. (I had to remove the cover and drag the thing over!) These problems were traced to an incorrect resistor value

on the (also cloned) controller board. Someone does not know the colour codes as it was off by a factor of ten. Moral — buy brand name.

Following transmittal of my comments to him, Strachan mailed me a (cloned?) Z-80 card. As with all material of this type, I have donated it to the Trinity Western College Computer Science department. After all, a journalist cannot profit directly from these reviews, but the College may as well get such fallout.

I will continue to evaluate workalikes as they come my way, but will only publish the results if the design is a significantly new variation on the theme — and on that note, let me pass something along to manufacturers.

Hello out There

My office is literally overflowing with material for this space, for my other review columns, and contract evaluations. If you have something you want me to take a look at, please tell me whether you want a review for publication or privately on a beta test or contract basis.

If the last of these three, please telephone or write to arrange terms before you send your product. You should include clear instructions as to disposition once the review is done. Lacking these, I will donate it to Trinity if it is of no immediate use to me.

Call -A.P.P.L.E. has its own setup for reviews, and I would never see anything sent to their office, as I am neither their employee nor agent, but a freelancer. ("Staff writer" just means "regular contributor.") So if you have anything to say or to send me, please communicate directly (address below with all the rest). So far, I have been able to respond to all letters or calls, but I cannot guarantee that this will always be possible.

Will the Real Dealer Please Stand Up?

You find out about some interesting situations when you do a little Northern Spying around town. A few weeks ago, I was in Vernon, B.C. and visited the two local stores which advertised Apples for sale. They turned out to be right across the street from each other.

Under the sign of a big Apple rainbow, I entered Rick's Compu-equip. Werner Schaeffer was on duty and proved to be a gracious and knowledgeable host to this visitor from the far off rain forests of the coast. He explained that this was the "only unauthorized Apple store in Canada" and that they were transshipping the machines from another dealer on the coast.

When I wondered how they could get away with this, he explained that it was being done with the knowledge and consent of Apple Canada, who were not anxious to shut them down, as they were exclusively Apple and were doing a good job at it. Negotiations for an official dealership had stalled due to the fact that the store across the street already had such a dealership, even though they were not exclusively Apple, but sold other machines as well.

The store was obviously well-equipped in every way to sell and service Apples, and I left feeling good about the whole operation, taking my traveling road show to the other side to the Dynamic Computer Centre.

Here, as expected, there were only two Apples on display, and most of the space was taken up by other brands. Fair enough, this was a general computer store, not a specifically Apple one. There also appeared to be a programming service being undertaken in the back, so this was clearly also a diversified organization.

However, the obviously inexperienced clerk on duty spent almost the entire time I was there helping a group of young boys boot up games on a VIC-20. These were not customers (well, perhaps I wasn't either), and in my view, this sort of thing is very detrimental to those who are. She did not introduce herself or provide a card, and seemed quite unsure of herself. Several questions got answers like "that has to come from our Kelowna store."

As I said, they carried the Apple, but there was no brand commitment like there was at the first store. If there is to be only one dealership on that street, I know who I would give it to, don't you?

END

Enough, enough, I cry. In the queue for coming months are discussions of some expensive add-ons, like $300 keyboards and 132 column boards, and some talks with Nellie about my new school for the educational section. Someday, I'll tell you what I think the ideal micro should be like. See you next time.

Suppliers and Publishers mentioned this month:

for CN! and ETI
Electronics Today International
Unit 6, 25 Overlea Blvd.
Toronto, Ontario, Canada
M4H 1B1

for 6502 boards, kits, and peripherals:
Exeltronics
319 College St.,
Toronto, Ontario, Canada
M5T 1S2

Surplustronics
3120 College St.,
Toronto, Ontario, Canada
M5T 1S3

(There are many, many others)

InfoAge
211 Consumers Rd., Suite 302
Willowdale, Ontario.
M2J 4G8

Compuwest
20 - 1152 Mainland St.,
Vancouver, B.C. Canada
V6B 2T9

Windfall
Data Base Publications Ltd.,
Europa House, 68 Chester Rd.,
Hazel Grove, Stockport
England
SK7 5NY

The author can be reached at:
Arjay Enterprises
Box 1253
Aldergrove, B.C.
Canada
V0X 1Z0

or

Trinity Western College
7600 Glover Rd.,
Langley, B.C.
Canada
V3A 4R9

READER SERVICE CARD

Free Information on the items circled

Card expires March 31, 1984 (1-84)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210

1. What kind of product(s) are you likely to buy in the next three months?

A. ☐ Hard Disk
B. ☐ Increased Capacity Floppy Disk
C. ☐ Apple Compatible Floppy Disk
D. ☐ Computer
E. ☐ Printer
F. ☐ Modem
G. ☐ CRT

2. More information on Apple PugetSound Program Library Exchange? A. ☐ Yes B. ☐ No

Name _____

Company _____

Address _____ City _____

State _____ Zip _____ Phone _____

MORE INFORMATION about the products and services in this issue FREE!

Just circle the number and fill out the card. (Note: card expires three months following month of publication.)

Act Now!

Join!

A.P.P.L.E.

Apple **P**ugetSound **P**rogram **L**ibrary **E**xchange

The **World's** Largest Apple Computer User Group

Be a part of the over 20,000 A.P.P.L.E. members and you'll receive:

- 12 issues of **Call —A.P.P.L.E.** magazine. The nation's premier Apple computer publication.
- Substantially reduced prices on sophisticated software, hardware, publications, and services.
- Access to the A.P.P.L.E. closed user group on the **SOURCE.** Excellent for information exchange and program downloading.
- **Call —A.P.P.L.E.** Hotline.

ALL THESE BENEFITS AND MORE FOR ONLY $51!

Detach and mail the Business Reply Application Form. →

Call —A.P.P.L.E. Reader's Choice

Help us help you. By filling out and returning this card, marked with your preferences, you can guide us in selecting material tailored to your needs. Just place the numbers from 1 to 10, in order of preference, in the appropriate boxes. Please do not choose more than 10 features. A.P.P.L.E. members who include their membership number, will have their names placed in a drawing for $100 credit toward any A.P.P.L.E. product. Drawing will be held March 20, 1984. The top scoring author will also receive a $100 cash bonus.

☐ 1 **Earl of Varspace** Cornelis Bongers
☐ 2 **The Festive Apple** Burt Webb
☐ 3 **Computerizing Science Exhibits** Dave Taylor
☐ 4 **Higher Text in Action** Steve Brugger
☐ 5 **"C" What I Mean?** Jim Hockenhull
☐ 6 **Data Analysis (Part 2)** Jeffrey Jacques
☐ 7 **Strictly Business** Mike Regeimbal
☐ 8 **Flying High...With E-Z Pilot** Earl Keyser
☐ 9 **Mind Storms** Eric Flescher
☐ 10 **Facemaker** Garry Margolis
☐ 11 **Telecommunications Conference** Michael Thyng
☐ 12 **1983 Index** Kathryn Suther/Val J. Golding
☐ 13 **The Gaming Room** Val J. Golding
☐ 14 **Koala Pad** Alan R. Fischer
☐ 15 **ProDOS and Friends** Cecil Fretwell
☐ 16 **A Canadian Harvest** Rick Sutcliffe
☐ 17 **Poet-tree** Michael Shadick/Sallie Stephenson
☐ 18 **A.P.P.L.E. Doctor** Bruce F. Field
☐ 19 **Write —A.P.P.L.E.** the readers
☐ 20 **Bytes From A.P.P.L.E.** Val J. Golding
☐ 21 **Editor Bytes Back** the editor
☐ 22 **So What Did You Expect?**

Name _____ Address _____ Memb. # _____ — _____

A.P.P.L.E. employees ineligible for prizes. Void where prohibited by law.

☐ Please send me an A.P.P.L.E. Apple-cation and information.
☐ $51.00 payment enclosed.
☐ Mastercharge/Visa

$25.00 First time fee
26.00 First year dues
$51.00 Total

MasterCard # _____ Exp. Date _____

VISA # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____

Zip _____ Phone _____

A.P.P.L.E., 21246 68th Ave. S., Kent, WA 98032
(206) 872-2245

First Class mail for **Call —A.P.P.L.E.** add $15. Otherwise third class mailed free as part of membership in U.S.A.

Additional Airmail postage required for foreign memberships and shipments.

Additional membership postage:

- Canada, West Indies, Central America, Columbia, Venezuela and Mexico — Add $15.00 ($60.00 total)
- All S. American countries (except above), Europe, and N. Africa — Add $21.00 ($66.00 total)
- Asia, Mid East, S. Africa, Australia and Pacific Islands — Add $27.00 ($72.00 total)

All orders must be prepaid in U.S. funds only.

Apple **P**ugetSound **P**rogram **L**ibrary **E**xchange
The WORLD'S largest Apple Computer user group

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 217 CLINTON, IA
POSTAGE WILL BE PAID BY ADDRESSEE

Call —A.P.P.L.E.
P.O. Box 1993
Clinton, IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA
POSTAGE WILL BE PAID BY ADDRESSEE

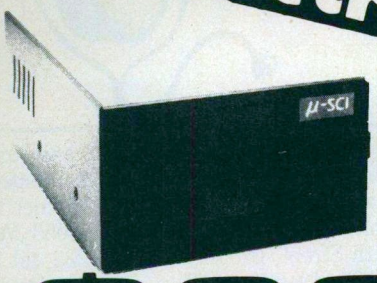
Call —A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA
POSTAGE WILL BE PAID BY ADDRESSEE

A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

Laing Electronics



μ-SCI

MICRO-SCI
A2 DRIVE

$229

BUY THE FIRST AND THE BEST APPLE COMPATIBLE DRIVE!

A2 DRIVE WITH CONTROLLER	$309

(FREE SHIPPING IN USA OR CANADA.
A2 MODELS ONLY.)

A40 (40 TRACK DRIVE W/O CONTROLLER)	$284
A70 (70 TRACK DRIVE W/O CONTROLLER)	$365
— CONTROLLER FOR A40 OR A70 DRIVES	$ 79
A3 APPLE III DIRECT REPLACEMENT	$299

10382 Stanford Ave.,E-203, Garden Grove, Cal., 92640
Phone (714) 534-4216

Check or Money Order. — NO CREDIT CARDS! Check's Allow 10 BANKING DAYS FOR CLEARANCE. Prices and specifications subject to change without notice. Add $7.50 for UPS surface. Cal. Res. add 6% Sales Tax.

Circle 8 on Reader Service Card

The Easy Way To Plan Great Dinners

Let us send you our exciting meal planning system. Try it for 2 weeks free with no cost or obligation.

Because our ad manager has a small weakness for pizza, we call it The Pizza Program. Actually, it's a complete meal planning system. It generates delicious dinner menus and shopping lists according to your tastes, your diet, and your budget.

It is a great time saver for anyone who cooks. You can quickly print out a new menu or shopping list for a day, a week, or any period up to 42 days at a time. It can even remind you when it's time to go out to your favorite restaurant. Plus, it can arrange your shopping list in sequence according to the isles at your local store.

Accept our 2 week free trial. There's no need to send any money now. Just send the coupon. We'll bill you later. If you're not satisfied for any reason, just return it and write cancel on the invoice. What could be more fair?

Gourmet Software

Gourmet Software, Dept. CA-10
3583 Barley Ct., San Jose, CA 95127

OK, Rush me The Pizza Program to try for 2 weeks and bill me later for just $34.50 plus $2 shipping. (Sales tax added in California). I understand I can return it within 21 days if not satisfied and owe nothing. My PC is an ☐ Apple II Plus or IIe ☐ IBM PC or XT ☐ Other _____ (Needs to run Apple or IBM software).*

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE ( ) _____ In case we have a problem with your order and need to call you.

P.S. For faster service call our ad manager, Rich Smith at (408) 866-0887.

*Apple and IBM are registered trademarks of Apple Computer and International Business Machines

Circle 13 on Reader Service Card

+

C

+

UNIX

AZTEC C65

A POWERFUL, PROFESSIONAL, AND PORTABLE "C" LANGUAGE DEVELOPMENT SYSTEM FOR APPLE COMPUTERS

AZTEC C65 brings all the power of "C" to the APPLE // and APPLE //e. The AZTEC C65 "C" compiler generates fast 6502 assembler code. Run time support includes full UNIX IO, general utility functions, scientific math functions, overlay support, and device driver interfaces.

AZTEC C65 is a professional development system. It includes APPLE SHELL — a UNIX like development environment, VED — a full screen editor, AS65 — a relocating 6502 assembler, LN65 — a full feature linkage editor, MKLIB — a library utility, and a number of other development utilities.

AZTEC C65 code is portable to AZTEC C compilers for CP/M-80, PC DOS, MS DOS, CP/M-86, TRS 80 and COMMODORE 64. Code is easily transferred to and from UNIX v7. Cross compilers are available from UNIX, PC DOS, MS DOS, CP/M-86, CP/M-80, and others.

PRICES

AZTEC C65 - APPLE DOS 3.3	$199
PC DOS, MS DOS, CP/M-86, CP/M-80 TO 6502 CROSS	$750
UNIX TO 6502 CROSS (PDP-11)	$2000
"C" TUTORIAL SYSTEM (upgradable to full AZTEC C65)	$95
AZTEC C65 MANUAL	$35
AZTEC C65 without manual	$169
AZTEC CII for CP/M-80 (MP/M-80)	$199
AZTEC C86 for PC DOS, MS DOS, or CP/M-86	$249

MANX
software systems

Call for information on AZTEC C65 for ProDOS and other 6502 systems.

ORDER BY PHONE OR MAIL — DEALER INQUIRIES INVITED

MANX SOFTWARE SYSTEMS
BOX 55, SHREWSBURY, NJ 07701
Order Phone: 800-221-0440
Information and NJ Orders: 201-780-4004 **Telex: 4995812**

Circle 14 on Reader Service Card

ACCESS ALL MICRO

11306 Southland Road
Forest Park, Ohio 45240

TOTAL ACCESS TO

Software

- BUSINESS
- EDUCATIONAL
- GRAPHICS/ UTILITIES
- PERSONAL PLANNING
- WORD PROCESSING
- BOOKS
- ENTERTAINMENT

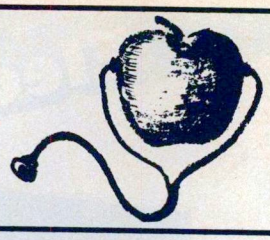
Hardware and Accessories

- PRINTERS
- DISK DRIVES
- INTERFACES
- MONITORS
- MODEMS
- DISKS
- RIBBONS

MasterCard VISA CALL
1-800-543-1114
Ohio · Alaska · Hawaii
Call 513-825-5803
CINCINNATI CUSTOMERS USE
OUR SOFTWARE PICK-UP FACILITY

Circle 1 on Reader Service Card

A.P.P.L.E. DOCTOR



Bruce F. Field

GEORGE Kinal has some additional comments on one of my answers in last month's column. The question was on low-cost modem kits. The questioner wanted to know if you could use the game-I/O port to drive a modem. George says it is possible to use the game port for simultaneous send and receive at 300 baud. Bizcomp and MFJ sell software for their modems which does exactly that.

Also, XICOM (801 E. Ogden, Naperville, IL) sells a bare board communications card for $18.

I have received some fan (?) mail on how to do super and subscripts with Wordstar and the C.Itoh printer from Albert Feldman and Arnold Rosenberg. I pass along their observations.

> The first thing is to select the line feed printer choice in Wordstar's Install program. This selection would ordinarily cause the C.Itoh printer to print double spaced. However, by installing a command for half-line feed in User Patch 3 and invoking that command at the beginning of the document, the superscript and subscript commands in Wordstar will work perfectly.
>
> So, using the Install Program, enter N for NO when you are asked, "Are the modifications now complete?" and type the following hex numbers exactly in the locations shown:
>
> **06D3 - 04**
> **06D4 - 1B**
> **06D5 - 54**
> **06D6 - 31**
> **06D7 - 32**
>
> This corresponds to ESC T12 which is the C.Itoh printer command for setting the printer to half-line spacing (12/144").
>
> Now, to return to regular spacing, type the following numbers in, which correspond to ESC A:
>
> **06D8 - 02**
> **06D9 - 1B**
> **06DA - 41**
>
> When using Wordstar, type [Ctrl-PE] at the beginning of your document to invoke the half-line spacing. Then, you may type regularly and get the superscripts and subscripts by using the Wordstar commands.
>
> If you want to get back to regular spacing, which, of course, will be double spacing, type [Ctrl-PR].
>
> Thanks, Albert and Arnold.

I have also received some mail on a question a few months ago regarding radiation from TVs or monitors upsetting nearby disk drives. The problem in that case was that disks initialized on his machine refused to boot on his machine, however, disks initialized on other machines would boot on his machine. Jeff (no last name) sent me the following.

> I built a set of wood shelves to go over my Apple and hold my disk drive and TV on top. I had no problem until I bought a second disk drive. It seems that the new drive with short cable and shielding was more sensitive to external RF radiation than the old drive! I solved the problem by simply placing a sheet of aluminum foil over the disk drive. No problems since. Some of my friends have had similar problems and since it is common practice to place a TV over the drives, this may happen frequently. Your readers may be interested in the simple solution.

Question:

Apple Inc. is upgrading rev. A motherboards in the //e for 560x192 pixel resolution free of charge. Note that if value is inversely proportional to publicity, this must be a gold mine. Is there some procedure for upgrading?

Answer:

First, 560x192 graphics resolution is only available if you have an 80-column extended memory card plugged into the auxiliary slot. This is the card that has an additional 64K of memory on it. As I understand it, Apple (through its dealers) will replace the rev. A motherboard for a rev. B *if* you purchase or have purchased the 80-column extended memory card (presumably from Apple).

It is not possible to use this extra resolution directly from Applesoft, special machine language routines are needed. Software Development Inc. (2053 West 11th Street, Brooklyn, NY 11223, tel. no. (212) 449-6300) sells a program Doublestuff, $39, for double Low-Res and High-Res routines callable from Applesoft.

Question:

Please explain the use of the ampersand to access subroutines. How would you program this in Applesoft?

Answer:

The ampersand is a reserved word in Applesoft (i.e. a command) that allows you to run machine language programs from Applesoft. When Applesoft executes the ampersand it jumps to a special memory location that tells it where to find the machine language program to be run. In order to use the ampersand you must have first loaded in a machine language program and set the special memory locations ($3F5,3F6) to point to the address of the machine language program.

All is not lost if programming in machine code is not one of your greatest joys in life. There are a number of commercially available programs that use the ampersand to perform special functions, usually formatted printing, inputting strings with commas, sorting, etc. A rather complete description of a number of ampersand utilities appeared in the March 1983 issue of *Call -A.P.P.L.E.* and a large portion of All About Applesoft (issue 1 of *Call -A.P.P.L.E. In Depth*) is devoted to ampersand utilities and Applesoft machine language routines.

letters

Question:

I would like to put a new volume number on my existing disks other than 254. I have used a disk zap utility to change track 17, sector 0, byte 6 as indicated in the DOS manual. However, upon rebooting, the same volume number 254 appears on the disk. How can I permanently change the volume number without re-initializing?

Answer:

When DOS initializes a diskette it writes an address field and a data field on the diskette for each sector. The address field contains the volume number of the diskette, the track, the sector, and a checksum. Disk zap programs are set up only to read the data field, the address field cannot be modified. The volume number in the address field is the one that DOS uses so that even if you change the one in the Volume Table Of Contents (track 17, sector 0) you still get the old volume number. I suppose it is possible to rewrite just the address field, however, I don't know of any program to do this. The closest I can come is to use the program in Bag of Tricks (Quality Software, Reseda, CA) to reinitialize individual tracks.

A time consuming but effective method would be to use FID from your system master diskette to copy all the files over to a newly initialized diskette. The wildcard character (=) may be used instead of typing all the file names. I don't necessarily recommend this because I would be afraid that an error might be introduced during the copy process. I may be paranoid, but I avoid unnecessary copying. If you had a program that could compare both disks after copying, then maybe...

Question:

Where can I get my Apple II+ upgraded to a IIe?

Answer:

There is no policy for upgrading II+'s to *IIe*'s. If you went to your dealer, they could replace the motherboard, keyboard, and case, probably for more than the cost of a new *IIe*. The best solution is to sell your II+ and just buy a *IIe*. If recent speculation holds, by January 1984 or so the price of the *IIe* may be reduced. A company (Computer Classified Bluebook, P.O. Box 3395, Reno, Nevada 89505) has compiled a Bluebook with the current selling prices of used computers, analogous to the Bluebook for used cars. At $85 a year it may be a bit expensive for an individual, but maybe a club could swing it.

Question:

If we buy a computer and TV set in this country and take it to another country where the power supply is 220 V and 50 Hz (cycles per second), will it work if we use a 220/110 V transformer? Even though we convert the voltage, what about the frequency, does it need to be converted?

Answer:

The Apple is not frequency sensitive and will happily work on 110 V at 50 Hz. TVs and monitors may not be quite so forgiving. Most TVs are rated for 60 Hz but may work on 50 Hz although the picture may shrink in size or be wavy. However foreign TV signals are different than those used in the U.S. so the TV can't be used to receive foreign TV programs. You may be able to find a monitor in the U.S. that is rated for 50 or 60 Hz.

If you go to Europe or some other place where U.S. military troops are stationed, you will find TV sets able to receive NTSC (U.S. type), PAL, or SEACAM signals. These sets should work with your Apple, be usable as a TV overseas, and still be useful when you return home.

Tom Riley points out that if you buy a 220/110 volt converter it should be a transformer type. Some of the cheaper converters use silicon controlled rectifiers (SCRs) and produce a non-sinusoidal waveform. The Apple will probably work with this, but TVs or monitors may have problems. Transformer type voltage converters are readily available overseas. Make sure you get one with a large enough power rating to handle the Apple, monitor, and whatever else you are going to use. The power consumption of the Apple depends on what you have plugged into it, a maximum value is probably less than 100 watts. Newer monitors or small screen TVs use about 30 to 75 watts.

Question:

I've just noticed something about my Apple II+ with two disk drives (Apple Disk II and Micro Sci with an Apple controller). When I turn off my Apple, the "IN USE" lights on the front of the drives flicker on briefly. Does this imply impending disaster?

Answer:

Not really. When the power is turned off to the Apple, all the voltages in all parts of the circuit do not immediately go to zero. Capacitors in different parts of the circuit maintain some parts of the circuit on momentarily longer and peculiar things can happen. This is the case with your flickering LEDs, and does not mean that anything is happening to the disks.

I don't know if Apple has ever published a comment on whether you should leave your diskettes in the drive when you turn the power off, but I think you should take them out first. There is no guarantee that when you turn the power off a stray signal to the read/write head won't destroy some data on the diskette. To prevent this possibility from happening, I always open the disk drive doors and usually remove the diskette before turning the power off.

In a way, the same concern applies when you turn the power on. Before the Apple really "gets going," strange things could happen. I don't think this is a major worry as thousands of Apple owners turn their computers on with disks in the drive without any problems. I have been told that users would have even fewer problems with booting disks if they let the drive start turning before closing the door on the diskette. This apparently helps in centering the diskette on the spindle.

Question:

Creative Computing, Oct. 83 p. 283 mentions that the Apple II motherboard has undergone revisions through IFR Rev. D. Can you briefly inform me on what they accomplished?

Answer:

I believe that basically these revisions were to make the motherboard somewhat cheaper or easier to manufacture. The original rev. 0 board only had two colors (green and violet, in addition to black and white). Rev. 1 and all subsequent boards could produce four colors (green, violet, blue, and orange). Early motherboards had RAM configuration blocks that could be strapped to allow use of 4K dynamic RAM chips or the "newer 16K RAM chips." When memory prices came down, the configuration blocks were eliminated. (Imagine a 4K Apple!) Rev. 7 and later Apples use a 2716 compatible ROM for the video character generator. Thus, modifying rev. 7 Apples to display lower case involves replacing only one chip. Earlier Apples require a small plug-in circuit board. Later revisions were also made to reduce the amount of RF interference generated by the Apple.

NIBBLE EXPRESS III
YOUR APPLE* WILL LOVE IT!

TABLE OF CONTENTS

Turtle Graphics 8
Applesoft Line Cruncher 14
TRAC Income System 19
MAMA 25
Amper Free Space Catalog 26
Machine Language Editor 27
Apple CALENDAR 36
Peaceful Coexistence 43
Apple Slugger 43
Reverse Key 45
Amper Find 45
Bond Manager 47
DOS 3 + 2 53
Amper Speed 57
Apple Darts 61
MicroCalc 63
Variable Cruncher 72
Life 77
Apple Flash 80
Checker 81
GO- Greeting Program 83
Quick Sort 86
Applesoft Line Editor 90
Othello 96
SPRINT 99
Disk Dump 102
Hi-Res Colors 103
Apple Record Command System 104
Apple Art Gallery 114
Game I/O Expmt 119
Super Keypad 123
Disk Commander 125
Compare Applesoft 132
Amp-L-Soft 136
Electronic Message Center 152
The Shape 155
Disk Map 159
Apple Bowl Football 161
Apple Scroller 167
Lower Case Letters 168
Recipe Box 170
Disk Doctor 178
Quasar II 182
DOS Command Entry 189
MLE Driver 189

NOW AVAILABLE!

NIBBLE EXPRESS is an Anthology of the Major Articles and Programs appearing in Volume 3 of NIBBLE Magazine . . . nearly $400 worth of Programs for your Apple II, ACE, or other Applesoft-compatible computer!

NIBBLE EXPRESS III contains *up-to-date enhancements, KeyPerfect Tables,* and the *Best of NIBBLE!*

Even if you have all 8 issues of NIBBLE in 1982, you'll want the Express to have updated program listings in one convenient package! It's a MUST for your Library!

NIBBLE EXPRESS III is a bargain at $17.95 plus $1.75 Postage/Handling ($2.50 outside the U.S.).

YOUR APPLE WILL LOVE IT!
(And so will you!)

NIBBLE
P.O. Box 325
Lincoln, MA 01773

Yes! I want to reserve NIBBLE EXPRESS Vol. III for my Library! Here's my ☐ Check ☐ Money Order for $17.95 plus $1.75 postage/handling. (Outside U.S. add $2.75 for postage/handling).

☐ Also send me NIBBLE EXPRESS Vol. I at $12.95 Plus postage/handling (see above).

☐ Send me NIBBLE EXPRESS VOL. II at $14.95 plus postage/handling (see above).

Master Card & Visa Accepted

Card # _____ Expires: _____

Signature: _____

Telephone: _____

Please Print Clearly

Name: _____

Address: _____

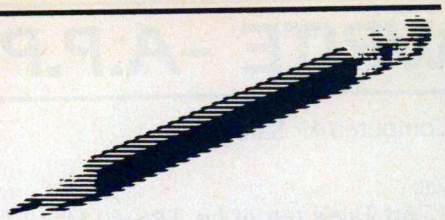
City _____ State _____ Zip _____

Your check or money order must accompany your order to qualify. Outside U.S.: Checks must be drawn on a U.S. Bank.

*Apple is a registered trademark of Apple Computer Inc.
*ACE is a trademark of Franklin Computer Co.

Circle 37 on Reader Service Card

WRITE —A.P.P.L.E.



the readers

Sweet Service

Dear Editor:

BRAVO to Clay Ruth for the excellent and understandable essay on a *Mockingboard Speech Editor* in the October issue.

Until that issue arrived, I thought that I was perhaps the only A.P.P.L.E. member who had plunked down their $199 for the Mockingboard special offer, as the device never surfaced again in any *Call -A.P.P.L.E.* print.

As a terrible typist, and assuming that there are more than just the two of us now owning Mockingboards among the membership, would it be possible to offer Clay's disk for sale as an aid in utilizing the device? Roughly how many of them were sold to members while the offer lasted?

In correspondence with *Sweet Micro* seeking some more polished documentation than the mimeo stapled pages sent initially, I learned that some of the "phonetic tricks" needed to overcome unexpected pronunciation may be due to either a faulty or poorly-chosen chip in the A.P.P.L.E. supplied versions. If you have a *Mockingboard* and the second chip from the left in the top row is a #4049, this is the wrong chip. It should be replaced with a #4050, which *Sweet Micro* supplied gratis and promptly along with some later-version documentation. When replacing, be very sure that the new chip is oriented properly! The notch *must* go to the right when the connector strip is at the bottom.

Note: Sweet Micro told me to replace the chip with their *new* #4050 even if the chip was already a #4050 in there.

Hoping other Mockingboarders will chirp-in comments.

Richard Scott
Arctic Agency
Drawer 80845 College Station
Fairbanks, Alaska 99708

Our Anthology diskette 10-B contains the Mockingboard Speech Editor program.

Plaudits for THE Spreadsheet

Dear Editor:

A member of the APPLE MOUSEE (our Fairbanks User's Group) mentioned that any VisiCalc-produced templates could be run on A.P.P.L.E.'s THE Spreadsheet. ("VisiCalc," a product of VisiCorp.)

With some experimenting, I found that templates could indeed be run, edited, converted to upper and lower case, etc. by using THE Spreadsheet's many features. This and more:

For those who use favorite spreadsheet templates many times per day or week, THE Spreadsheet offers a major convenience aside from any other features. It is distributed on the "honor system," i.e., it can be backed-up, deleted, or configured as you wish. So, it is comparatively simple to take a backup copy (!), delete the demo files and unused 80-column/70-column drivers, and store numerous templates—plus THE Spreadsheet itself— right on a single disk. If an 80-column board is present, an added convenience is one-disk fast loading directly into an 80-column display of the template, ready to use.

Also, for members owning VisiCalc's older Apple][+ version which comes up "VC-193..." when "/V" is typed, a $20 update to the "VC-202..." version is worth it. A letter to the company stating the present version and serial number will bring the details.

Richard Scott
P.O. Box 80845
College Station
Fairbanks, AK 99708

A Piece of PI

Dear Val,

In the April issue the article "Trig, Degrees/Common Logs" the constants are (as I've written before) not converted to the best value by the Applesoft interpreter. Use of PI = 3.14159265, RA = 57.295779505 will guarantee that the converted values in hex most closely agree with the correct value. Because of the Applesoft conversion algorithm, the best value to input is not necessarily the best decimal approximation. Instead of using "1.5708," a better practice would be to first define, say, P2 = .5*PI and then use P2 in the DEF FN. More accurate and faster. Also, if you like the practice of multiplying instead of dividing as one of your earlier articles suggested (!) for speed, the code " /RA " should be substituted by " *DG " where DG = .01745329252.

These numbers were tested by conversion by the interpreter and subsequent inspection of the hex representation in the variable as stored by Applesoft. The number of digits are, I think, the minimum you can use to insure all 32 bits of the FP mantissa are correct.

In closing, let me recommend to interested readers the DTACK-Grounded MC68000 board advertised in recent issues of your magazine and MICRO. I have also owned the Arithmetic Processor for Apple and even for that one application — using the board to do FP operations in Applesoft — the DTACK board is significantly better. And that application taps only a tiny fraction of its potential. It is well made and rugged, has given me nearly a year of trouble free operation and has been very easy to write software for with the Pase Zero Assembler. Incidentally, I use Big Mac as an Editor and find it superior to the Toolkit and perfectly compatible with the 68000 assembler.

During development of a random number generator for the 68000, I wrote a routine that generated the numbers and; after performing several tests, stored them into several hundred bins. Ten million 32-bit numbers generated and tested takes less than half an hour. In the realtime graphics demo supplied with the board that generates 12 frames of hi-res per second, most of the time the 68000 is in idle while the Apple clears the screen. With block transfer coming — well you get the idea — anyway if you like that sort of thing, get one or start saving for it.

Peter W. Soule
969 Via Del Monte
Palos Verdes Estates, CA 90274

letters

WRITE -A.P.P.L.E.

Computer/Air Safety

Val,

Art Ude's test of his TRS-80 Model 100 for electronic interference on airplanes was not adequate to prove their safety. He proved that his particular computer did not cause problems in that particular airplane when its radios were tuned to whatever frequencies he happened to be using that day.

You can't interpret that to mean that several people operating a combination of computers and games in a 300 passenger airliner won't cause problems. You can't even be sure this wouldn't cause problems under different circumstances.

Even if no one stops you, it wouldn't be wise to operate one during an instrument approach.

W.C. McFarlin
245 SW 298'th Place
Federal Way, WA 98003

CP/Muffin

Dear Val,

In the March issue of *Call -A.P.P.L.E.*, I was pleasantly surprised to find your CP/MUFFIN Routine for transferring TEXT files from the CP/M environment to Apple DOS. I hope there will be more forthcoming for the Apple-Z80 environment.

I am enclosing a copy of Microsoft Application Notes for the Apple Z80 Softcard. Note the routine on page 4 - CPMXFER. Unlike yours, the routine is initiated in the CP/M environment and then transferred to Apple DOS where it resides to make the actual transfers from CP/M to DOS. Also, note that it transfers both .DOC (Text, ASCII, etc) and .COM (obj) files. I find the program very useful for exchanging routines with friends, in both Apple DOS and CP/M. One Apple DOS disk can have Applesoft, Integer and CP/M files. The CP/M files are transferred from DOS back to CP/M using the APDOS Utility on the Softcard System Master.

Though CPMXFER does not have the optional ability to copy an entire disk, as your CP/MUFFIN, it does have the nice feature of being able to Catalog disks of both environments once loaded i.e.,: Drive 1(A!) defaults to Apple DOS and Drive 2(B!) defaults to CP/M. Typing the commands CAT or DIR will Catalog the selected format.

I don't think this information is proprietary in that the covering letter from Microsoft states it is non-copyrighted material and may be freely exchanged by Softcard users. Since I am a registered owner of Softcard CP/M and also a member of A.P.P.L.E., I feel that all bases are covered. Also, note that these Application Notes are for the 2.20 (56K) version of Softcard CP/M. I understand that some of the patches are not valid for the newly released version 2.23 (60K). Hope this information is of interest to you and really appreciate the endeavor that you and the A.P.P.L.E. staff are making.

H.W. Madison
105 Crestline Drive
Bellingham, WA 98226

Extended Text Card Problems

Dear Val,

I have discovered two new, undocumented features (read bugs) in the Apple *IIe* 80 column board. These "features" can be a significant problem for the programmer who is not aware of them, and the Apple Technical Support people say to "program around it." Both are related to the use of the arrow keys in GET statements when in 80 column mode.

The first involves GETting A$ with a right arrow which results in A$ holding the ASCII value of 21 [Ctrl-U]. But in 80 column mode, the value of A$ is whatever the cursor was resting on on the screen. If the cursor was on a blank, the ASCII value of A$ will be 32 (space). If some other character, that value will be assigned to A$. In my application, I was trying to use the arrow keys to move around the screen between possible entry locations, or enter a value if a number key was struck.

The second problem encountered involves the up arrow key (ASCII 11 [Ctrl-K]). When using Apple's 80 column board, the board intercepts all ESC characters to switch the 80 column cards edit mode. But if a RETURN is struck after a series of up arrows, the card not only exits escape mode, but clears the screen below the cursor. If you are trying to move around on a formatted screen, an accidental ESC can be a disaster.

Ordinarily, these matters would not be a problem. After all, most of us are pretty good typists, or at least slow enough to correct our mistakes before it is too late. But this application is for my fellow workers who are intimidated by bank machines. For them, one keystroke can be very traumatic if the screen goes away.

If anyone has a suggestion on how to avoid these features, I'm ready to try it.

Denis Anson, M.S., OTR
2010 N.W. Kearney St.
Portland, OR 97209

Frank Comments on Apple Dot Matrix Printer

Dear Colleagues:

C.ITOH = APPLE DMP = PROWRITER?

Just a follow up on *Apple Doctor* (August *Call -A.P.P.L.E.)* and the letter from Jerry Battenhouse (November *Call -A.P.P.L.E.).* I work at a State Institution. We recently purchased an Apple *IIe* and Apple dot matrix printer. Actually, the purchase order specified the *IIe* and a Prowriter Printer, *or equivalent.* The reason for this specification was that I knew how well my Prowriter at home (purchased from A.P.P.L.E.) worked with graphics and text; especially technical manuscripts with sub- and superscripts that were prepared with ZARDAX. We accepted the Apple DMP as an equivalent of the Prowriter. Imagine the lost hours (what are we doing wrong?.. are you sure the DIP switches are right?, etc., etc.) and then shock and dismay when we found out that *the Apple DMP is not equivalent to the Prowriter!* Thanks to a well-known local expert who saved us from more wasted time. He told us that the Apple DMP won't handle any sub- and superscripts.

The local Apple dealer is making a good faith effort to get my problem resolved, but until things change fundamentally, I suggest that you see all of the features you need in a printer in order to operate on a truly equivalent system. Those of us in institutions have to put things out for bid. To paraphrase a well-known astronaut about to be blown into space, we "keep thinking that the whole thing has been put together by the lowest bidder."

Frank F. Vincenzi
3205 109th S.E.
Bellevue, WA 98004

letters

POET-TREE

Michael Shadick/Sallie Stephenson

MY MICRO MOM!

'Tis said Mom's pushing seventy
(She doesn't show her age).
A secret to her youth? She's now
A home computing sage!
Her microsystem's spread throughout
Her roomy mobile home
— Which makes it quite convenient
Wherever she may roam!

The keyboard's in the kitchen,
Which she calls her "micro center."
She likes to have it handy, for
New recipes to enter.
She keeps 'em all on BASIC files,
With programs she has written.
She calls up tasty menus, when
The cooking bug has bitten!

Among the sev'ral monitors
My Mother's home does boast,
Is one that's in the living room.
For Mom's a gracious host.
Displayed upon that monitor,
When dinner guests arrive:
The evening's menu. How this helps
Their appetites to thrive!

The breakfast nook is cozy, warm.
It also boasts a printer!
The bedroom was it's home — but then,
Mom had to be a sprinter,
And rush from "micro center" to
The "printout center." Now,
It only takes a step or two.
'Tis niftier? And how!

Of what does Mom make printouts? Well,
I thought you'd never ask!
She's found a way to expedite
Most any micro task!
Indeed, she's started publishing
A little printed "weekly."
"It's all the news you'd ever want
To know," My mom says meekly.

By now, you've guessed — my Mom's computer
Hobbies mean a lot!
She says they make her feel just like
A geriatric tot!
Mom's won a lot of converts to
Her brand of home computing.
"I don't have time to age," says she.
"Too busy with recruiting!"

RAISING AND LOWERING THE MIGHTIEST OF RIVERS — WITH AN APPLE!

Of twenty-six great locks and dams
Controlling Old Man River,
St. Anthony's the mightiest.
Its size makes small boats shiver!
The Mississippi tends to be
Too shallow, on his own;
Each lock and dam can hold him back,
Until his depths have grown.

You ask: What nautical-type chores
The Apple has to do?
St. Anthony must rise and fall,
To let the vessels through.
The Army Corps of Engineers
(Which runs all dams and locks)
Has Apple][‘s to help keep boats
From running on the rocks!

The river's flow is monitored
With Apple CP/M.
As flood gates open, close, or hold,
The river's tide they stem.
And meanwhile, boats are lining up
Upon the Apple's screen.
So when it's time for 'locking through,'
Their signal lights turn green.

The Master of the Locks controls,
From high atop his tower,
The closings and the openings
Of valves with massive power.
Beside him, is a console which
Has earned much lockside fame.
It's small, compared to mighty dams.
But, Anthony is its name!

A RATHER SALTY APPLE

To those of us computerists
On solid terra firma,
The Mississippi River is
As alien as Burma!
For Old Man River's atmosphere
Is not conducive to
Computing. Or, just MIGHT it be?
We'll see, before we're through!

Our mission! to investigate
The muddy Mississip,
In terms of what computing might be
Done at any slip.
A 'slip,' in case you don't know,
Is where a boat's moored to.
(We didn't know that term, ourselves;
We're landlubbers, it's true!)

Undaunted by our total lack
Of nautical-type knowledge,
We nonetheless are launching forth
A (barely) floating college!
Aboard our '50s house boat is
An ancient Apple I.
The micro's nickname: 'Salty.'
(It has yet to come undone!)

We've chosen this companion on
Our sail to New Orleans,
To serve us as a Second Mate
That also cooks and cleans!
You think that we exaggerate
Our Salty's sailing talents?
She'll make our ship's log entries — while
She keeps our (floating) balance!

Before we're through, we hope to find
Some great computing on
The Mightiest of Rivers! With
The floods and floatsam gone,
We pray we'll find smooth sailing as
The Mardi Gras draws nigh.
We're trusting in ol' Salty's help,
To keep us high and dry!

UNCLASSIFIEDS

Call -A.P.P.L.E. (un)Classifieds

Members Non-commercial rate .05/word
Members commercial rate .25/word
All others: 1.00/word, $25 minimum
Send ad copy with payment to editorial office. Unclassified copy must be received by the first day of the month preceding the date of issue.

MICROMOVER quickly transfers files between Apple disks over a phone line, without conversion first. Data Compression algorithm. DOS 3.2 and 3.3. Hayes Micromodem II and acoustic modems accommodated. Only one copy required, as it transfers itself to distant end. Cost is $50.00 from DAB Systems, 9116 Rockefeller Lane, Springfield, VA 22153.

ALMOST WHOLESALE — Save BIG BUCKS on almost all Hardware & Software needs. Usually 20% to 45% below retail. Monthly specials. Send self-addressed stamped envelope for latest price list, or contact us to request quotes on specific items. COMPUTER PRODUCTS UNLIMITED, Apple Dept., 11116 East 46th Ave., Spokane, WA 99206. (509) 922-1346 (eves). CompuServe (73075,613). SOURCE (TCG249).

AIR NAV WORKSHOP][+ /IIe], fully-customizable, navigate any area, adjustable winds, automatic flight path replay, much more. $40. Check, MC/V, UPS-COD. SPACE-TIME ASSOCIATES, 2039-CA Country Club Drive, Manchester, NH 03102, (603) 625-1094, Free Info.

COMPUTER ROAD ATLAS - You enter departing city and destination city. ROADSEARCH computes the shortest practical route — and more. Prints a detailed route with miles, time and fuel. Contains 406 USA/Canada cities and 70,000 road miles. DOS 3.3. 15 day moneyback guarantee. $34.95 (plus $1.50 S/H) Check/VISA/MC. At your dealer or Columbia Software, Box 2235R, Columbia, MD 21045, (301) 997-3100.

COMPLETE ADALAB PACKAGE by Interactive Microware Inc. Includes interface card, 16 channel fast A/D convertor. Software includes Lab Data Manager, Vidichart, Quickersample, Stripcharter, Quick I/O, & AI13 Data Acquisition Module. All documentation included. Hardly used, like new. $995. Dennis Mercadal, 1151 Cheshire, Casper, WY 82609, (307) 235-4949.

APPLE PUBLIC DOMAIN SOFTWARE FOR SALE. Seven Different Disk Volumes. Volume 1-7 for $12.00 each disk including shipping. All seven disks for $50.00. $1.00 for a catalog of all disk volumes. Send money order or check to: Gennaro Conte, 2522 Holland Ave., Bronx, NY 10467. Compuserve ID: 74005,1064 Delphi user name: CONTE or call me at (212) 798-2964.

NON-FLASHING CURSOR — The original NFC for Apple][/][+ has been enhanced & improved. Menu driven installation results in intelligent NON-FLASHING CURSOR. Complete documentation and theory of operation included on disk. Support for PLE/GPLE, Diversi and David DOS also built in. Accept no substitutes! Send $9.95 check or MO + $1.00 shipping to: RER Software, 630 Riverview Lane, APPLEton, WI 54915.

APPLE SHAPE EDITOR — Everything you ever really wanted (or needed) but couldn't buy this cheap! Why spend a bundle for a user's NIGHTMARE when there's a powerful and user-friendly editor, shuffler, compressor, appender, merger, and displayer with ONLINE HELP for only $19.95? Clean up tables created by expensive programs! Includes popular type fonts and complete documentation. Apple][+ /IIe DOS 3.3. Send check or M.O. today. Ask for MIRACLEWORKER. COMPUTER ODYSSEY, Box 10434, Pittsburgh, Pennsylvania, 15234. PA residents add 6%.

RENT SOFTWARE $5 MONTHLY. No membership fee. Send SASE to: Preowned Software, 4229 Baker Ave NW, Seattle, WA 98107 for list.

DISCAT SAVES YOU TIME AND MONEY. It's an Apple-compatible menu-driven catalog on diskette for one-step software shopping. The catalog contains a listing of over 750 items — at significant savings to you. Software is listed by name, publisher, type, or price range, or you can browse leisurely through the whole catalog at your own pace. Hurry! Gain access by botting up with DISCAT. Send $3 (Maryland residents add 5% state sales tax) to: Computer Ware Unlimited, Dept. CA; P.O. Box 1247; Columbia, MD 21044.

ENGINEERS, SCIENTISTS, and ALL Apple][+ /IIe users: Function-plotting program plots any portion of any function in CARTESIAN or POLAR coordinates. Many options; prompt-driven; simple. Dump Hi-Res plots on printer. Fully copyable. Written by CalTech Ph.D. Manual with mathematical hints included. Only $29.50. Peter Hohenbrink, P.O. Box 866, Placentia, CA 92670

MULTI-LINGUAL SOFTWARE for Spanish, French, German and other languages. Apple][+ or IIe. FREE CATALOG. Le Professeur, P.O. Box 301A, Swanton, Vermont, 05488, (514) 747-9130.

WANTED: Lisa-assembly programmers to test subroutine library and library maintenance program. Subroutines include: Input, Output, String, DOS, Math and Logical Transfer routines. Unique address structure simplifies coding of large machine language programs. Send name, address, system description and brief statement of Lisa Assembler experience to: M.A. IANNCE, 1009 Country Hills, Santa Maria, CA 93455

APPLE BUFFS! Moving, and must clear out the following interesting material: Lot 1: Original (1977) Apple][Reference Manual - precedes Red Book. Lot 2: The January 1978 Red book. Lot 3: Original Applesoft Reference Manual (November 1977, blue cover, large format). Original Applesoft Quick Reference Guide (1977, blue cover). Lot 4: Apple Contact complete (issues 1-6, 1978-1979). Lot 5: Call -A.P.P.L.E., Sep-Dec 1978, 1979-1983 complete. Lot 6: Apple Orchard, 1980-1983 complete. Lot 7: Micro 6502, "Best" Vol 1, Oct-Dec 1978, 1979-1980 complete, 9 scattered 1981-1982. Lot 8: Wozpak I (well used) and Wozpak II (books only for both). I also have Byte, Kilobaud/Microcomputing and Creative Computing issues 1977-1983. Call for details. Will accept best offer for each lot within 30 days from this ad. All prices FOB my home in New Jersey. Duncan Bailey, (201) 666-3992.

SKYFORTH: COMPILED LANGUAGE SYSTEM—Skyforth has over 300 commands with floating point, 32-bit integers, print formatting, Hi-Res graphics, screen editor, sorting and memory search and move. Advanced features include program overlays and chaining, and color fill. The disk also includes an assembler, many utilities and a turnkey system generator. The system uses standard binary and text files but also includes commands for easy access to individual and ranges of sectors. Fully compiled for ultra-fast execution. Rusns the BYTE PRIME number benchmark in 160 seconds compared to about 200-250 seconds for other FORTH languages and Pascal and over 2800 seconds for Applesoft. Diversi-DOS compatible. Requires 64K Apple][/][+ /IIe and DOS 3.3. Versions for Arithmetic Card or Corvus available. 110-page manual included. Price $95 (WA add 7.9% tax). TOSCH Information Management, 3711 SW 107th ST., Seattle, WA 98146. (206) 246-3839. "Unlike some other versions of FORTH, Skyforth fits neatly into the Apple operating system, making a combination that has the best features of both." Softalk, October 1983.

IDS 440G Printer w/cable $235. Versawriter-like Digitizer $80. Dynatyper typewriter interface $90. Call Tom Tosch (206) 246-3839, 3711 SW 107th St. Seattle, WA 98146.

VIDEX VIDEOTERM 80 column card with soft switch and Inverse video chip. duplicate order by mistake, box has never been opened. $240. R.A. Goebel, 6105 Horton Ave., Shreveport, LA 71105.

STOLEN: Apple][+ Computer (serial # 543349), AMDEK Color I monitor (serial #Y2K042567), two 3.3 D.O.S. (serial #'s: 850240 and 1021930) and fan. If anyone has information about whereabouts of this equipment, please contact Washington State Dairy Council, at 632-9335 in Seattle.

SUBLOGIC GRAPHICS MADE EASY! Use BASIC to create 3D color graphics and animation quickly and easily. Dust off your SubLOGIC disks (or get them from us) - but forget the hexadecimal math, byte-swapping, peeking and pokeing. Use METALOGIC's powerful set of "&" commands to program your own graphics applications without ever leaving the BASIC language. Detailed manual not only teaches you to use metalogic, but includes documented sample programs and an advanced applications section. SubLOGIC A2-3D2, ROM Applesoft, RAM card and DOS 3.3 required. METALOGIC only $50 or $155 including A2-3D2. Send check or M.O. to Artemis Systems, Inc., 513 West Mt. Pleasant Avenue, Livingston, NJ 07039. NJ residents add 6% sales tax.

LETTER QUALITY PRINTER AT LOW COST!! Used Diablo 1620 daisy wheel printing terminal has true proportional spacing, underlining, boldface, and can double as a printing terminal or even a typewriter. Printing speed up to 45cps and bi-directional printing is possible. Wide carriage with serial cable to connect to your serial card or modem. With new manual, ribbon and printwheel- $425. This baby is no toy and the results are beautiful! Bruce Amrose, Box 71006, Los Angeles, CA 90071. Work (213) 628-0131.

FOR SALE, WORDSTAR 3.30 and also PIE WRITER II, never used or even opened, $225 and $63, respectively, Ralph Smith, day: (714) 632-3024, night: 544-4586, 13472 Flint Drive, Santa Ana, CA 92705.

APPLE WRAPPERS. You were smart enough to buy the best computer on the market. Don't let your investment sit unprotected. Buy the BEST COVER on the market. Made of BEAUTIFUL VELOUR that will protect your investment for years to come. Vinyl lined keyboard. Designer colors: Light Gold, Dark Blue, Tobacco. Also available in Blue Denim. Keyboard $14.50, with Monitor /// $19.95, single drive $8.50, stacked drives (2) $11.50. Shiping $1.50 each. Printer covers and other configurations available by special order. SATISFACTION GUARANTEED!! A/PEEL, 2990 Barbara St. Ashland, OR 97520.

Call —A.P.P.L.E.

MONTHLY MEMBER SPECIAL

- Be the "first on your block" with a ProDOS utility.

ProZap $22.50

INTRODUCTORY PRICE

Product Code — SPZAP

- Reads, edits and writes any ProDOS block or DOS 3.3 sector.
- Includes PRODOS. SYSTEM and BASIC. SYSTEM as SYS files.

Note: Included documentation covers ProZap ONLY, not ProDOS.

- On the same disk: Monitor utility examines, searches memory for hex, ASCII.

Offer good til 3/31/84

shipping weight 2 oz

24 hour toll-free order no.

800-426-3667

(except in Washington)
Washington state residents add 8.1% sales tax

A.P.P.L.E. Orders
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

Predictable.

A flawless future is in sight with 3M diskettes.

When it comes to keeping track of precious data, predictable means reliable. Being able to count on every diskette, every time. At 3M, reliability is built into every diskette. We've been in the computer media business for over 30 years. And we've never settled in. We're constantly improving and perfecting our product line, from computer tape and data cartridges to floppy disks.

3M diskettes are made at 3M. That way, we have complete control over the entire manufacturing process. And you can have complete confidence in the reliability of every 3M diskette you buy.

Look in the Yellow Pages under Computer Supplies and Parts for the 3M distributor nearest you. In Canada, write 3M Canada, Inc., London, Ontario. If it's worth remembering, it's worth 3M diskettes.

3M hears you...

Circle 25 on Reader Service Card

3M

REMEMBER.

ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison

Circle 10 on Reader Service Card